

1

AD-A263 756



Research Product 93-07

DTIC
ELECTE
MAY 4 1993
S c D

Concept Formulation Process Aid for Trade-Off Determination

Operator's Guide

93 5 03 018

93-09366



73pg

March 1993

STRICOM, Orlando Field Unit, Florida
Training Systems Research Division

U.S. Army Research Institute for the Behavioral and Social Sciences

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction
of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Acting Director

Research accomplished under contract
for the Department of the Army

Human Resources Research Organization

Technical review by

J. Douglas Dressel

Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

DTIC QUALITY INSPECTED 8

NOTICES

FINAL DISPOSITION: This Research Product may be destroyed when it is no longer needed.
Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: This Research Product is not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 1993, March		3. REPORT TYPE AND DATES COVERED Final Apr 91 - Sep 92	
4. TITLE AND SUBTITLE Concept Formulation Process Aid for Trade-Off Determination: Operator's Guide				5. FUNDING NUMBERS DAHC35-89-D-0046 63007A 795 3104 C05	
6. AUTHOR(S) Page, Gary L.; Blacksten, H. Ric; Elder, B. Leon; Sticha, Paul J. (HumRRO); and Singer, Michael J. (ARI)					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Human Resources Research Organization 66 Canal Center Plaza Suite 400 Alexandria, VA 22314				8. PERFORMING ORGANIZATION REPORT NUMBER --	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Institute for the Behavioral and Social Sciences ATTN: PERI-IF 5001 Eisenhower Avenue Alexandria, VA 22333				10. SPONSORING/MONITORING AGENCY REPORT NUMBER ARI Research Product 93-07	
11. SUPPLEMENTARY NOTES Contracting Officer's Representative, Michael J. Singer					
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				12b. DISTRIBUTION CODE --	
13. ABSTRACT (Maximum 200 words) This report describes the structure of the prototype concept formulation process aid for trade-off determination. The goal was to adapt previously developed models for use during a specific step in the development of a training device. This report identifies the hardware requirements and the necessary configuration and describes the installation procedures. A brief overview of the aiding system execution and use is presented. The report also describes the structure and function of the program files. The data elements, links between elements, and aiding functions are described in the context of the program routines.					
14. SUBJECT TERMS Training system optimization Training system models Medium selection				15. NUMBER OF PAGES 72	
				16. PRICE CODE --	
17. SECURITY CLASSIFICATION OF REPORT Unclassified		18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified		19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	
				20. LIMITATION OF ABSTRACT Unlimited	

Research Product 93-07

Concept Formulation Process Aid for Trade-Off Determination: Operator's Guide

Gary L. Page, H. Ric Blacksten, B. Leon Elder, and Paul J. Sticha
Human Resources Research Organization

and

Michael J. Singer
U.S. Army Research Institute

STRICOM, Orlando Field Unit, Florida
Stephen Goldberg, Chief

Training Systems Research Division
Jack H. Hiller, Director

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

March 1993

Army Project Number
2Q263007A795

Training Simulation

Approved for public release; distribution is unlimited.

FOREWORD

A major problem faced by the military is determining how much simulation is necessary to reach the stated learning objectives. Behavioral and analytical techniques to assist in rapid determination of required features for training devices exist, but they are infrequently used. In addition, information on the cost-effective use of training devices in courses of instruction is sparse.

The U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) and the Simulation, Training, and Instrumentation Command (STRICOM, previously the Project Manager for Training Devices PM TRADE) joined efforts to investigate and develop models, databases, and analytical techniques that could support the design of advanced training technology (Memorandum of Understanding on Technical Coordination, May 1983; MOU Establishing ARI Field Unit, Mar 1985; PM TRADE and ARI expanded MOU, July 1986). The potential effect on the Army will be to reduce the cost of fielded training devices and, at the same time increase their instructional effectiveness. The ARI work was performed under the task entitled Advanced Technology for the Design of Training Devices and Simulators. The prototype concept formulation process aid (CFP-Aid) described in this report will aid engineers in organizing and using training requirements information to produce an initial set of training device configurations for analysis.

STRICOM has maintained partnership in the development and evaluation of the CFP-Aid prototype. The models and techniques in the prototype provide a basis for supporting the integration of behavioral and engineering data, knowledge, and expertise in training device design. This product provides sufficient documentation on system requirements, system structure, and program files for maintenance of the prototype software. Final product and user evaluation briefings were held in December 1991 and July 1992, respectively. Managers from STRICOM's Research and Engineering Management Division participated in both briefings. STRICOM management is considering options for application and further development.



EDGAR M. JOHNSON
Acting Director

CONCEPT FORMULATION PROCESS AID FOR TRADE-OFF DETERMINATION:
OPERATOR'S GUIDE

CONTENTS

	Page
INTRODUCTION	1
GETTING STARTED	2
Hardware Requirements	2
Configuration Requirements	2
Installation	3
Starting the Program	3
User Interaction	3
OVERALL SYSTEM STRUCTURE	4
Conceptual System Structure	4
General Program Structure	5
Program Files	6
System Files	9
Working Files	10
Pointers	10
File Structures	11
DESCRIPTION OF PROGRAM FILES	11
Program Initiation	12
Module 0--File Management	12
Module 1--Requirements	14
Module 2--Components	20
Module 3--Analyses	33
Module 4--Documentation	38
INDEX	41
REFERENCES	43
APPENDIX A. CFP-AID INTERPRETER PROGRAM FILES	A-1
B. CFP-AID DATABASE FILE STRUCTURES AND INDEX FILE KEYS	B-1
C. RULE BASE DEVELOPMENT GUIDANCE	C-1

CONTENTS (Continued)

Page

LIST OF TABLES

Table 1. File Naming Guidelines	7
---	---

CONCEPT FORMULATION PROCESS AID FOR TRADE-OFF
DETERMINATION: OPERATOR'S GUIDE

Introduction

The Concept Formulation Process Aid (CFP-Aid) software is a prototype decision aid designed to be used in the Trade-Off Determination (TOD) Phase of the Concept Formulation Process (CFP) in the training device development process. The aid is designed to assist the designer in proceeding systematically through the elements of the trade-off determination (Elder, Sticha, Page, & Singer, in press; Sticha, Gibbons, & Singer, in press). The CFP-Aid was designed to aid the user in selecting training requirements; examining training requirements; identifying required instructional features and fidelity levels; conducting cost, effectiveness, risk, and schedule (CERS) analysis; and documenting applied requirements and analysis results.

CFP-Aid helps the user select training requirements by selecting training requirements from a master list of functions; through the identification of training requirements based on the identified Military Occupational Specialty (MOS), Additional Skill Indicator (ASI) and Special Qualification Identifier (SQI); or by direct entry of required tasks or functions. The system helps organize and examine training requirements by allowing the user to specify and link requirement categories and cues to functions or tasks. The aid helps the user identify instructor support features and required fidelity levels for visual and motion characteristics through the use of expert system rules. The aid helps the user to evaluate component cost, effectiveness, risk, and schedule factors by providing spreadsheets and formulas and by providing tables of the analysis results. The documentation function provides a trace or audit trail of user selections and analysis results.

The purpose of this operator's guide is to provide sufficient guidance for minimal maintenance of the Concept Formulation Process Aid (CFP-Aid) prototype software. The level of guidance was selected because the software is in the prototype stage, meaning that true extensive use by large numbers of users is not possible. The next step in the software engineering development process would be to remove the CFP-Aid from the host environment, which would entail a rewrite of the software and make any extensive maintenance guidance moot. This step is required for the system to be more widely used, implemented on networks, expanded with additional training requirements information, and to provide a wider range of fidelity rules.

Any maintenance of the CFP-Aid as currently formulated requires considerable familiarity with the MS-DOS Single User GURU software Version 3.0 (Micro Data Base Systems, Inc., 1991a & b). No attempt has been made in this document to guide the reader in trouble-shooting the software or using GURU functions.

The first section of this guide provides instructions for installing, starting, and using the software. The second section provides an overview of the file types, the file naming conventions, and the structure of the data files. The last section provides brief descriptions of the functions performed by the program files and relationships between those files.

Getting Started

Hardware Requirements

The Concept Formulation Process Aid (CFP-Aid) software is designed to operate with the MS-DOS Single User GURU software Version 2.01 or 3.0 (MDBS, 1991a; 1991b) on the standard IBM PC/XT/AT or compatible computer with hard disk drive and EGA, VGA color, or monochrome monitor. The computer must have at least 640K bytes of RAM and 5 megabytes of room on the hard disk for the GURU software and full operation of associated programs. The CFP-Aid prototype operates within these GURU software memory requirements. The CFP-Aid software responds to keyboard input and to input from any GURU-compatible mouse having a minimum of two buttons.

Configuration Requirements

When the GURU software is installed on the computer the INSTALL procedure makes changes to the AUTOEXEC.BAT and CONFIG.SYS files in the root directory. The AUTOEXEC.BAT file is changed to include the PATH for the directory containing the GURU programs. CFP-Aid (based on GURU requirements; MDBS, 1991a, 1991b) requires space for at least 15 disk buffers and 50 files in the CONFIG.SYS file. The CONFIG.SYS file should therefore also be changed to include the following lines or values:

```
BUFFERS=15
FILES=50
```

If these changes are not made during installation of the GURU system, problems in operation could occur. If necessary, use DOS procedures to make the necessary changes to these files. Any changes that are made will not go into effect until the system is rebooted.

As with all database applications, third party disk caching software will significantly enhance the speed of operation of CFP-Aid. Disk caching refers to programs that use the memory (RAM) in larger systems that is not used by MS-DOS. The software uses some portion of that memory to emulate a rapid access memory storage device, which increases the amount of information that can be directly accessed by the program. This speeds operation of the software by decreasing the number of times the disk is used.

Installation

The CFP-Aid software comes on two 5-1/4" 1.2 Mb disks or two 3-1/2", 1.4 Mb disks. Use the following procedure to install the CFP-Aid on your computer:

1. Place disk 1 in a disk drive.
2. Make that drive the current drive by entering the drive letter and a colon. For example, if you put disk 1 in drive a:, you would enter the following:

a:<ENTER>

3. Start the install program by typing:

INSTALL d: directory<ENTER>

where, "d:" and "directory" are the identification of the drive and the name of the directory on which to install the CFP-Aid. Make sure that there is a space between the drive and the directory name. You must enter a drive letter, but the directory name is optional. If you do not enter a directory name, the software will be installed on a directory named \CFP. For example, if you enter

INSTALL c: \CFP-Aid<ENTER>

then the software will be installed on the C:\CFP-AID directory (creating the subdirectory, if necessary). If you enter

INSTALL d:<ENTER>

then the software will be installed on the D:\CFP directory. The program will prompt you to put in disk 2 when required. The install program will overwrite any existing files (with the same names) on the target directory.

Starting the Program

To begin program execution, change to the \CFP directory (or the directory on which you installed the CFP-Aid). Type the word GO, and press ENTER. The GURU logo will appear on the screen for a few seconds. The logo will disappear and the first CFP-Aid Menu will be displayed.

User Interaction

A mouse may be used to move through the menu selections and to move the cursor on the screen. The left button of the mouse is the ENTER function and is used to select items from the menu.

Clicking on a word in the main menu activates a pull-down menu. Clicking on an item in a pull-down menu displays an additional menu, list, graphic, or executes a process. The right button of the mouse is the "escape" function and is used to back out of menus and procedures.

Letters and numbers required as input by the user in response to screen prompts are entered through use of the keyboard. Movement through the menus can also be accomplished by using the arrow keys. When a menu item is highlighted, press ENTER and the system will perform the indicated function. Menu selections can also be made by typing the letter corresponding to the capitalized letter in the menu choice (ENTER is not required). The ESC key serves as the "escape" function key.

Overall System Structure

Conceptual System Structure

The CFP-Aid is a menu-driven system consisting of five modules. The main menu is displayed at the top of the screen (see below). Each use of the CFP-Aid program begins and ends with this menu. There are information windows embedded in the menu selections that allow operation of the modules. The main menu provides one-word descriptors for the five modules. For a new project the user will normally work in each of the modules, moving from left to right through the menu. Each of the five modules is briefly described in the order they appear in the menu.

Concept Formulation Process Aid				
File	Requirements	Components	Analyses	Documentation

File	The File module allows the user to specify the project to begin work on or select a project to continue work on. The File module allows an easy change from one project to another or to a different version of a project. It allows the user to save a copy of a project using a new name, delete a project from the system, print rule base data and data bases, and exit from the program.
Requirements	The Requirements module helps the user to define the goals of the training device being designed. It is used to identify tasks or functions that are to be trained on the device, to select the perceptual cues and response options that are required for training the tasks and functions, and to document the reasons for simulation.

Components	The Components module assists the user in specifying the Instructional Features and Fidelity required by the tasks/functions. Also, it allows the user to specify the system components for consideration.
Analyses	The Analyses module allows the user to compare the cost, effectiveness, technical risk and schedule (CERS) for specific components and systems based on user input. It also automatically generates systems from selected components.
Documentation	The Documentation module generates an outline that helps the user to document work with the CFP-Aid system and to provide information to the TOD.

General Program Structure

The CFP-Aid is a collection of different types of computer files that perform within the GURU host software system (MDBS, 1991a, 1991b). These include:

- Program files - these contain the CFP-Aid program code (instructions) for GURU;
- System data files - contain the "master" CFP-aid reference data;
- Working files - containing project-specific information.

GURU is an interpretive, as opposed to a compiled programming environment, therefore it is necessary to have the GURU software installed on the computer and executing as the host for the CFP-Aid.

CFP-Aid files were named by concatenating relevant file descriptors lexicographically, corresponding to a logical file hierarchy. File types are identified by their extensions, as required by GURU (see above). At the lowest level of the hierarchy, arabic serials are used to distinguish files. For example, OG05A1.IPF is the program file accomplishing the Print List_of_Projects function under the File menu, and was named based on the following logic:

- OG - Initial characters on all program files.
- 0 - Module 0 (File menu)
- 5 - Print Menu
- A - Print List of Projects
- 1 - first (and only, in this case) program involved
 in this service
- .IPF - standard extension for Guru interpreter program
 files.

Two additional constraints affect the naming conventions. First, in accordance with MS-DOS requirements, file names are limited to a maximum of eight alphanumeric characters, plus a three character (maximum) file extension. Consequently, file names are unavoidably cryptic. Second, although a hierarchical tree structure is used, the tree occasionally includes gaps where no program element now exists. This is largely a result of program evolution, as space was left for possible features, or as certain program elements were determined unnecessary. (Such gaps may actually prove convenient for later programming modification since fewer existing files must be renamed when inserting a new program file.)

Program Files

Program files instruct the GURU interpreter to perform the operations and calculations constituting CFP-Aid. Program files should be modified only with great care.

Every program file has the extension IPF (Interpreter Program File). Naming guides and program functions are listed in Table 1 for the top three menu levels. As an example, the program file used to accomplish automatic cue determination is the first program in process 1A (Automatic, under Requirement Categories) under module 1 (REQUIREMENT DETERMINATION MODULE); therefore the name of this program file is OG11A1.IPF.

Program files are all located in the main CFP-Aid directory, created and named during CFP-Aid installation.

Program files are ASCII text files, and may be viewed through use of the DOS TYPE command, or with any text editor. Each program file contains an introductory comment section describing the purpose of the program. Program files are described in more detail in Section III. A complete list of program files is given in Appendix A.

Table 1. File Naming Guidelines

FILE MODULE	0	
New	0	Start new project
Open	1	Resume existing project
Select Project	1A	Choose from list
Enter key phrase	1B	Select using description
Save as...	2	Save under new file name (copies files)
Project status	3	Display project status
Delete	4	Delete project(s) from list
Print	5	Print data
List projects	5A	Print list of projects/descriptions
Function databases	5B	Print function determination databases
Cue databases	5C	Print cue databases
Checklist	5D	Print simulation checklist results
IF rule data	5E	Print instructional feature rule base data
Fidelity rule data	5F	Print fidelity rule base data
Component databases	5G	Print component databases
Documentation databases	5H	Print documentation databases
All	5I	Print all databases
Exit	6	Quit (files automatically closed)
REQUIREMENTS DETERMINATION		
MODULE	1	
Required Functns.	0	Determine functions
Select from MOS	0A	Restrict choices (MOS, ASI, SQI)
Master list	0B	Use list to select functions
Review	0D	Review selections (can deselect)
Enter/edit	0E	Maintain working TASK/FUNCTION files
Requirement Cat.	1	Determine sources of cues
Automatic	1A	Use links from functions with review
Master list	1B	Use list to choose cue sources
Review	1D	Review results (can deselect)
Enter/edit	1E	Maintain master CUE data files
Simulation		
Checklist	2	Answer questions
Enter/edit data	3	Maintain master data files (alternate entry point) - MOST, ASI, SQI, TASK/FUNCTION, CUES

(Continued)

Table 1. File Naming Guidelines

(Continued)

COMPONENTS		
MODULE	2	
Inst. features	0	IF rulebase
Automatic	0A	Use results of functions
Auto. w/ confirm	0B	rulebase with user confirmation of data
Answer questions	0C	rulebase and user answers questions
Justify	0D	Review results (why)
View results	0E	Draw graph of results
Enter/edit	0F	Maintain Inst. Feature Name master files
Fidelity	1	fidelity rulebase
Automatic	1A	Use results of cue/functions
Auto. w/ confirm	1B	rulebase with user confirmation of data
Answer questions	1C	rulebase and user answers questions
Justify	1D	Review results (why)
View results	1E	Draw graph of results
Enter/edit	1F	Maintain Fidelity Dim. master files
Select Components	2	Select components
Automatic	2A	Use cue, fidelity, IF with review
Master list	2B	Use list to choose components
Review	2D	Review results
Enter/edit	2E	Maintain master Component files
Enter/edit	3	Maintain master data files; Instructional Feature, Fidelity, Components
ANALYSES MODULE 3		
Component CERS	0	Conduct components tradeoffs
Cost	0A	Compare component cost
Effectiveness	0B	Compare component effectiveness
Technological	0C	Compare component technical risk
Schedule	0D	Compare component schedule
Overall	0E	Make overall component comparison
Create systems	1	Create systems from components
Automatic	1A	Use results of components with review
Master list	1B	Use list to create systems
Review	1D	Review results (can deselect)
System CERS	2	Conduct system tradeoffs
R&D cost	2A	Compare system R&D cost
Effectiveness	2B	Compare system effectiveness
Technological	2C	Compare system technical risk
Schedule	2D	Compare system schedule
Overall	2E	Make overall system comparison
Lifecycle cost	2F	Compare system lifecycle cost
Enter/Edit	3	Maintain master system files

(Continued)

Table 1. File Naming Guidelines. (concluded)

DOCUMENTATION		
MODULE	4	
Generate outline operations	0	Generate outline from prior operations
Edit outline	2	Edit outline
Print	3	Print outline
Print to file	4	Prints outline as a text file

System Files

CFP-Aid system files are of four types:

- "Master" data files -- distinguished by form OG*.ITB (extension identifies table file)
- "Master" data file indexes -- distinguished by form OG*.IND (extension indicates index file)
- Rule bases -- distinguished by extension RSS (Rule Set Source file)
- "Compiled" rule bases -- distinguished by extension RSC (Rule Set Compiled file)

The "master" data files, and their associated index files, are used by the program during the conduct of CFP-Aid processes. These data files provide reference file structures, reference task/function data, MOSSs, cues, and pointers (see Appendix B). In addition, certain system files are temporary files used for memory or for intermediate results. The names of these files are constructed in analogy to the naming of program files, with the module identified that most importantly relates to that master or intermediate data file. The correct file extension (ITB) is added to give the complete program file and extension.

Remarks: (1) Not all ITB files are system files; some are working files, to be discussed next. (2) All system files are located in the main CFP-Aid directory, whereas all working files are located in subdirectories of the main directory. (3) Not all ITB files have associated index files. (4) A given master data file may be used in a number of different program files; thus, naming master data files can be somewhat arbitrary.

Three rule bases are provided. Others may be added (see Appendix C for guidance on rule base development). One is IFRB.RSS (and its compiled version IFRB.RSC), the Instructional

Feature Rule Base. The others are Fidelity rule bases: MOTION.RSS, for motion and seat movement rules; and VISUAL.RSS, for visual display characteristics rules; each with its respective compiled (*.RSC) version.

Working Files

Working files are the data files (and their associated index files) created for particular projects investigated by the CFP-Aid during employment of the CFP-Aid. These working files are located in subdirectories of the main CFP-Aid directory. CFP-Aid creates a subdirectory for each project entered by the user. The subdirectory name consists of the letters "CFP" followed by a two digit serial number, indicating the current project (up to 100 projects, each in its own subdirectory, numbered 00 to 99 can be accommodated). The first project is automatically created by CFP-Aid, assigned the project serial number 00, and appropriate files are placed in the created subdirectory CFP00. Subsequent projects are assigned sequentially increasing numbers and appropriate files are placed in the referenced subdirectory. The working files are also located in the associated project directory. Each file will have a name consisting of the project directory name, as a root, followed by a module number, function number, and serial number.

When a new user opens his or her first project, CFP-Aid creates subdirectory CFP00 to the main CFP-Directory, which is the first project subdirectory. Assuming CFP-Aid was installed in C:\CFP, the newly created working subdirectory would be C:\CFP\CFP00. As this user exercises CFP-Aid modules for this project, CFP-Aid creates or updates files in this subdirectory. Each file in the subdirectory begins with the subdirectory name and is followed by a three digit sequence for module + function + serial number. Thus, within the initial project (CFP00), when this user exercises module 3 (analyses), for function/process 1 (create system), working files are created named CFP00310.ITB, CFP00311.ITB, etc., together with their associated index files, as needed.

Pointers

CFP-Aid files generally contain information for mapping back and forth to predecessor and successor files. Thus, pointers are included in intermediate files. All except the outline are many-to-many links, while outline is one-to-many. The pointers establish links between the program elements:

```
functions --> outline sections
           --> cues
           --> instructional feature rule data
           --> fidelity rule data
           --> components
```

```

        cues --> outline sections
            --> instructional feature rule data
            --> fidelity rule data
            --> components
    checklist --> outline sections
instructional features --> outline sections
                    --> fidelity rule data
        fidelity items --> outline sections
                        --> components
                        --> components
        components --> outline sections
components CERS --> outline sections
    systems CERS --> outline sections

```

File Structures

Database (table) file structures and their index file keys are described in Appendix B. Index file keys are critical to the correct functioning of retrieval routines in GURU (MDBS, 1991a, 1991b). Many problems can arise when the index is missing or has been corrupted. Unfortunately, this can occasionally happen spontaneously with the GURU software. In addition, it is often difficult to determine that a data file has a corrupted index. The clearest indication is a system failure message that will appear with a referenced line number when running the CFP-Aid under the GURU debugger routines. Unfortunately, these will not be noticed during normal operations because GURU flags errors without interrupting execution of the program or displaying the error flags. When the system appears to lose data, or return nonsensical information, the problem probably lies in the index. The most direct way to assess this is through the use of the debugger routines in GURU. The simplest fix is to identify the database file being accessed or searched and use GURU functions to recreate a corrected index. This will require you to open or use the data file, "Mark" the records false to prevent erasure, "Compress" the database file to delete the corrupted index file, and "Index" the file using the appropriate field (identified in Appendix B). Complete information about the "Mark" and "Compress" commands may be found in the GURU reference manuals (MDBS, 1991a, 1991b). Unfortunately, extreme care must be taken when compressing or indexing files so that data is not inadvertently lost. If the corruption occurs in linking files, they should be printed out and inspected to insure that correct links are maintained between data files.

Description of Program Files

The following sections describe the functions of the program files. These program files include those developed by ARI to provide data entry and editing functions, as well as files developed by HumRRO that perform selection and analytical functions. The level of a program file in the hierarchy is

indicated by the naming conventions, discussed earlier. A summary listing of all program files is included as Appendix A.

Program Initiation

Program initiation is accomplished by the files shown by the following two Interpreter Program Files (*.IPF):

CFP-AID.IPF - show main menu. This is the top-level routine for CFP-Aid. It sets global variables and presents the main menu.

OGA.IPF - simulate main menu. This routine simulates the exercise of the top level menu. It is used to portray non-active menus on the screen as background for an active menu.

Module 0--File Management

File management is handled by the following program files:

OG0.IPF - display file pull-down menu. This routine displays and acts on the file pull-down menu for those services accessed from File in the main menu.

OG0A.IPF - simulate file pull-down menu. This routine simulates the exercise of the file menu.

OG001.IPF - create new project. This routine is the new project creation function. The routine creates a project file record and duplicates the OG400 outline file and its index as CFPNN400, where NN is the project serial number (this convention is described above, used throughout the report, and will not be presented again). Since the new project is assigned the next available serial number from the memory file (OG0A), this number is also updated.

OG01.IPF - display open menu. This routine displays the open menu.

OG01A1.IPF - project menu list. This routine presents a list of the projects in a menu. The only data file used here is OG0B (and its index), the project file.

OG01B1.IPF - select project. This routine allows the selection of projects by description. The only data file used here is OG0B (and its index), the project file.

OG021.IPF - save as.... This routine is the "save as..." project function. It creates a project file record and duplicates all existing project files (for the current project) as a new project. Since the new project is assigned the next

available serial number from the memory file (OG0A), this number is also updated.

OG031.IPF - project status. This routine shows the current status of the current project by showing which modules have been completed.

OG041.IPF - delete project. This routine is the delete project function in the file module. The only data file used here is OG0B (and its index), the project file.

OG05.IPF - display print menu. This routine displays the print menu.

OG05A1.IPF - print project list. This routine prints out a project list in the file/print module. This routine uses the file OG0B, the project file and its index.

OG05B1.IPF - print task/function database. This routine prints out a function list in the file/print module. The associated cues for each function, as well as associated components, are also printed.

OG05C1.IPF - print cue database. This routine prints out a cue list in the file/print module. The associated components for each cue are also printed.

OG05D1.IPF - print simulation checklist. This routine prints out the simulation checklist in the file/print module.

OG05E1.IPF - print instructional features data for functions. This routine prints out the instructional features rule base data in the file/print module.

OG05E2.IPF - print instructional features data for cues. This routine prints out the cue features from IF's in the file/print module (only called from OG05E1.IPF).

OG05F1.IPF - print fidelity data for functions. This routine prints out the fidelity rule base data in the file/print module.

OG05F2.IPF - print fidelity data for cues. This routine prints out the cue features from fidelity in the file/print module (only called from OG05F1.IPF).

OG05F3.IPF - print fidelity data for instructional features. This routine prints out the instructional features from fidelity in the file/print module (only called from OG05F1.IPF).

OG05G1.IPF - print component database. This routine prints out a component list in the in the file/print module.

OG05H1.IPF - print outline template. This routine, OG05H1.IPF, prints out an outline list in the file/print module.

OG05I1.IPF - print all. This routine calls each of the print routines in order and "prints all."

Module 1--Requirements

Program routines addressing training requirements aspects of training system design are described below:

OG1.IPF - display function pull-down menu. This routine presents the function menu.

OG1A.IPF - simulate function pull-down menu. This routine simulates the exercise of the functions menu.

OG10.IPF - display task/function determination menu. This routine presents the function determination menu.

OG10A1.IPF - MOS selection. This routine presents the MOS menu, allowing selection of the MOS in the function selection module. The only data file used here is OG100, the MOS database.

OG10A2.IPF - ASI selection. This routine presents the Additional Skill Index (ASI) menu in the function selection module. Selecting an ASI identifies related tasks/functions. This routine creates or adds to the work file CFPNN100.IPF. The working file contains duplicates of the records from the functions database. Its structure is identical to OG104. The data files used here are OG101 the ASI database, OG103 the function intermediate file, and OG104 the function database. (An intermediate file links elements in files.)

OG10A3.IPF - SQI selection. This routine presents the Skill Qualifications Index (SQI) menu for the function selection module. The SQIs are listed for user selection, which identifies related tasks/functions. This routine uses and adds to the work routine created by OG10A2.IPF, named CFPNN100.ITB. The working file contains duplicates of the desired records from the functions database. The structure is identical to that of OG104. The data files used here are OG102 the SQI database, OG103 the function intermediate file, and OG104 the functions database.

OG10A4.IPF - review selections. This routine presents the review functions menu portion of the function selection module. This routine uses the work routine created by OG10A2.IPF, named CFPNN100.ITB to present a list of the selected task/functions. The working file contains duplicates of the desired records from the functions database. Its structure is identical to that of OG104. The working file is the only one used in this routine.

OG10B1.IPF - task/function selection. This routine presents the menu for the function selection module. The list contains all tasks/functions in the master database and allows the user to select (or remove from consideration) the task/functions for available for inclusion in the analysis. This routine creates, if necessary, and uses a work file named CFPNN100.ITB. The working file contains duplicates of the desired records from the functions database. Its structure is identical to that of OG104. OG104, the functions database, and this working file are the only ones used in this routine.

OG10D1.IPF - review tasks/functions. This routine presents a menu listing the selected tasks/functions for review. This routine uses the work routine created by OG10A2.IPF, named CFPNN100.ITB. The working file contains duplicates of the desired records from the functions database. Its structure is identical to that of OG104. The working file is the only one used in this routine.

OG10E1.IPF - working task/function maintenance menu. This routine presents the maintenance menu for the working file task/function database (CFPNN104.ITB). The routine presents three editing options for selection: edit, add, and delete.

OG10E2.IPF - working task/function edit routine. This routine edits records in the working task/function database (CFPNN104.ITB). Users are prompted to select specific records to edit. The selected records are then presented in a function edit form, which permits the field variable of the record to be changed. Changes made to the working task/function database by this routine are not kept in the Master task/function database.

OG10E3.IPF - working task/function add routine. This routine presents a message about using the master edit routines to add task/functions to the database. The message points out that for trial runs, working files may be edited to test the data.

OG10E4.IPF - working task/function delete routine. This routine presents a message suggesting that the user deselect the task/function (use the selection routine, OG10B1.IPF, to remove the task/function from consideration) that they desire to delete, as a simpler alternative.

OG11.IPF - display cue source menu. This routine presents the cue source menu under the function menu selection from the file menu.

OG11A1.IPF - automatic cue source. This routine automatically uses cue links to identify the cues associated with the selected tasks/functions. This routine creates or uses the work routine CFPNN110.ITB. The working file contains duplicates

of the desired records from the cue database. Its structure is identical to that of OG111. The work file and the functions database (CFPNN100) are the only ones used in this routine.

OG11A2.IPF - update cue to function index. This file updates the cue to function backwards index.

OG11B1.IPF - cue selection. This routine presents the menu for the cue selection module. The list contains all cues, and allows the user to select (or deselect) the cues for inclusion in the analysis. This routine creates, if necessary, and uses the work routine named CFPNN110.ITB. The working file contains duplicates of the desired records from the cue database. Its structure is identical to that of OG111. OG111, the cue database, and this working file are the only ones used in this routine.

OG11D1.IPF - cue review. This routine presents a list of selected cues for review under the cue selection module. This routine uses the work routine created by OG11A1.IPF named CFPNN110.ITB. The working file contains duplicates of the desired records from the cue database. Its structure is identical to that of OG111. The working file is the only one used in this routine.

OG11E.IPF - cue maintenance menu. This routine acts as a branch to the maintenance routines in OG13E.IPF. That routine presents the maintenance menu for the master cue database (OG111.ITB). The routine presents three maintenance options for selection: edit, add, and delete.

OG121.IPF - simulation checklist. This is the simulation checklist routine. The routine uses or creates, if necessary, the file CFPNN120. The program uses the data to indicate the task/function and cue requirements for simulation. This file contains a copy of the selected records from OG120.IPF, the checklist database. These two files and the selected functions database CFPNN100 are the only ones used by this routine.

OG13.IPF - maintenance for requirement module. This routine presents the maintenance menus for the Requirements module. It allows the user to select master files for editing. The users can select and modify any of the following master file databases: MOS, ASI, SQI, task/function, and cues.

OG13A.IPF - MOS maintenance menu. This routine presents the maintenance menu for the master MOS database (OG100.ITB). The routine presents three options for selection: edit, add, and delete.

OG13A1.IPF - MOS edit routine. This routine edits selected records in the master MOS database (OG100.ITB). Users are

prompted to select specific records to edit. The selected records are then presented in a MOS edit form, which permits the field variables of the record to be changed. All changes made to the master MOS database by this routine are permanent.

OG13A2.IPF - MOS add routine. This routine adds records to the master MOS database (OG100.ITB). A MOS edit form is presented to the user so that field variables for the new MOS record can be entered. Error checking routines are included so that a newly added record does not conflict with an existing record in the database. The newly identified record is added to the master MOS database, and appropriate links to the ASI (OG101.ITB), SQI (OG102.ITB), and functions (OG103.ITB) intermediate files are created. All records created by this routine are permanently added to the master MOS database.

OG13A3.IPF - MOS delete routine. This routine deletes selected records from the master MOS database (OG100.ITB), and deletes associated records in the ASI file (OG101.ITB) and SQI file (OG102.ITB). All records deleted by this routine are permanently deleted from the master MOS database.

OG13B.IPF - ASI maintenance menu. This routine presents the maintenance menu for the master ASI database (OG101.ITB). The routine presents three options for selection: edit, add, and delete.

OG13B1.IPF - ASI edit routine. This routine edits records in the master ASI database (OG101.ITB). Users are prompted to select specific records to edit. The selected records are then presented in a ASI edit form, which permits the field variable of the record to be changed. All changes made to the master ASI database by this routine are permanent.

OG13B2.IPF - ASI add routine. This routine adds records to the master ASI database (OG101.ITB). An ASI edit form is presented to the user so that field variables for the new ASI record can be entered. Error checking routines are included so that a newly added record does not conflict with an existing record in the database. The newly identified record is added to the master ASI database, and appropriate links to the MOS (OG100.ITB), and functions (OG103.ITB) intermediate files are created. All records created by this routine are permanently added to the master ASI database.

OG13B3.IPF - ASI delete routine. This routine deletes selected records from the master ASI database (OG101.ITB), and deletes the links in the functions intermediate file (OG103.ITB). All records deleted by this routine are permanently deleted from the master ASI database.

OG13C.IPF - SQI maintenance menu. This routine presents the maintenance menu for the master SQI database (OG102.ITB). The routine presents three options for selection: edit, add, and delete.

OG13C1.IPF - SQI edit routine. This routine edits records in the master SQI database (OG102.ITB). Users are prompted to select specific records to edit. The selected records are then presented in a SQI edit form, which permits the field variable of the record to be changed. All changes made to the master SQI database by this routine are permanent.

OG13C2.IPF - SQI add routine. This routine adds records to the master SQI database (OG102.ITB). An SQI edit form is presented to the user so that field variables for the new SQI record can be entered. Error checking routines are included so that a newly added record does not conflict with an existing record in the database. The newly identified record is added to the master SQI database, and appropriate links to the functions (OG103.ITB) intermediate file are created. All records created by this routine are permanently added to the master SQI database.

OG13C3.IPF - SQI delete routine. This routine deletes selected records from the master SQI database (OG102.ITB), and deletes the links in the functions (OG103.ITB) intermediate file. All records deleted by this routine are permanently deleted from the master SQI database.

OG13D.IPF - task/function maintenance menu. This routine presents the maintenance menu for the master task/function database (OG104.ITB). The routine presents three options for selection: edit, add, and delete.

OG13D1.IPF - task/function edit routine. This routine edits records in the master task/function database (OG104.ITB). Users are prompted to select specific records to edit. The selected records are then presented in a function edit form, which permits the field variable of the record to be changed. All changes made to the master task/function database by this routine are permanent.

OG13D2.IPF - task/function add routine. This routine adds records to the master task/function database (OG104.ITB). A function edit form is presented to the user so that field variables for the new function record can be entered. Error checking routines are included so that a newly added record does not conflict with an existing record in the database. The newly identified record is added to the master function database, and appropriate links to the fidelity intermediate (OG210.ITB), function intermediate (OG103.ITB), cue intermediate (OG110.ITB), component intermediate (OG220.ITB), and instructional features intermediate (OG200.ITB) files are created. Records are also

added to the Instructional Features database (OG202.ITB) and Fidelity database (OG212.ITB). The data in these files can be entered upon creation, or edited later. If data is not input, the "unknown" value code is entered for all variables. All records created by this routine are permanently added to the master task/function database.

OG13D3.IPF - task/function delete routine. This routine deletes selected records from the master task/function database (OG104.ITB), and deletes all associated links in the functions intermediate file (OG103.ITB), the cue intermediate file (OG110.ITB), the component intermediate file (OG220.ITB), the Instructional Feature intermediate file (OG200.ITB), the Fidelity intermediate file (OG210.ITB). It then deletes the associated records in the Instructional Feature database (OG202.ITB) and the Fidelity database (OG212.ITB). The records in the functions intermediate file (OG103.ITB) linking the to-be-deleted task/function to ASIs and SQIs are altered to "blank" values. All records deleted by this routine are permanently deleted from the master task/function database.

OG13E.IPF - cue maintenance menu. This routine presents the maintenance menu for the master cue database (OG111.ITB). The routine presents three options for selection: edit, add, and delete.

OG13E1.IPF - cue edit routine. This routine edits records in the master cues database (OG111.ITB). Users are prompted to select specific records to edit. The selected records are then presented in a cue edit form, which permits the field variable of the record to be changed. All records changed by this routine are permanent changes to the master cue database.

OG13E2.IPF - cue add routine. This routine adds records to the master cue database (OG111.ITB). A cue edit form is presented to the user so that field variables for the new record can be entered. Error checking routines are included so that a newly added record does not conflict with an existing record in the database. The newly identified record is added to the master cue database, and user-selected links to the fidelity intermediate (OG210.ITB), component intermediate (OG220.ITB), and instructional features intermediate (OG200.ITB) files are created. The routine creates new associated records in the Instructional Feature database (OG202.ITB) and the Fidelity database (OG212.ITB). The user is allowed to enter data at creation or leave the data elements marked as unknown. All records created for the master cue database, the Instructional Feature database, and the Fidelity database by this routine are permanently added.

OG13E3.IPF - cue delete routine. This routine deletes selected records from the master cue database (OG111.ITB), and

deletes the links in the instructional feature intermediate file (OG200.ITB), the fidelity intermediate file (OG210.ITB), and the component intermediate file (OG220.ITB). The associated data records in the Instructional Feature database (OG202.ITB) and the Fidelity database (OG212.ITB) are also deleted. The records in the cue intermediate file (OG110.ITB) linking the to-be-deleted cue to functions are altered to "blank" values. All records deleted by this routine are permanently deleted from the master databases.

Module 2--Components

Program routines addressing device component characterization are described below:

OG2.IPF - display component pull-down menu. This routine presents the Components menu.

OG2A.IPF - simulate component pull-down menu. This routine simulates the exercise of the components menu.

OG20.IPF - display instructional feature menu. This routine displays the instructional feature menu.

OG20A1.IPF - automatic instructional feature executive. This routine executes the instructional feature rule base using the previous module's results and asking all appropriate questions necessary to arrive at adequate answers. The routine creates, if necessary, and uses work files named CFPNN200.ITB, CFPNN201, CFPNN202, CFPNN203 and CFPNN204. The CFPNN200 file is an intermediate file to working copies of the instructional feature data. Its structure is identical to that of OG200. The CFPNN201 file is a working copy of the instructional features intermediate file. Its structure is identical to that of OG200. CFPNN202 contains a working copy of the instructional feature data for each function or cue. CFPNN203 contains the selected instructional features. Its structure is identical to that of OG203. CFPNN204 contains the backwards links from instructional features to functions and cues. Its structure is identical to OG0C. The instructional features intermediate file (OG200.ITB), the instructional features data dictionary file (OG201.ITB), the instructional features database (OG202.ITB), the instructional features data file (OG203.ITB), and the working files are the only ones used in this routine.

OG20A2.IPF - review rule base variables and execute IF rule base for selected cues. This routine executes the instructional feature rule base for each cue selected previously. In addition to the files opened in OG20A1, the working cue file, CFPNN110 is used.

OG20A2A.IPF - create and update selected IF and intermediate files. This routine creates and/or updates the selected instructional feature file as well as the selected instructional feature intermediate file. It is intended to be executed only from OG20A2 as it requires files opened in that routine as well as variables established there.

OG20A2B.IPF - create and update selected IF backwards files. This routine creates and/or updates the backwards selected instructional feature intermediate file. It is intended to be executed only from OG20A2 as it requires files opened in that routine as well as variables established there.

OG20A3.IPF - review rule base variables and execute IF rule base for selected tasks. This routine executes the instructional feature rule base for each function selected previously. In addition to the files opened in OG20A1, the working function file CFPNN100 is used.

OG20A3A.IPF - create and update selected IF and intermediate files. This routine creates and/or updates the selected instructional feature file as well as the selected instructional feature intermediate file. It is intended to be executed only from OG20A3 as it requires files opened in that routine as well as variables established there.

OG20A3B.IPF - create and update selected IF backwards files. This routine creates and/or updates the backwards selected instructional feature intermediate file. It is intended to be executed only from OG20A3 as it requires files opened in that routine as well as variables established there.

OG20B1.IPF - automatic, with confirmation, instructional feature executive. This routine executes the instructional feature rule base by ignoring the previous module's results and asking all appropriate questions necessary to arrive at adequate answers. The routine creates, if necessary, and uses work files CFPNN200.ITB, CFPNN201, CFPNN202, CFPNN203, and CFPNN204. The CFPNN200 file is an intermediate file to working copies of the instructional feature data. Its structure is identical to that of OG200. The CFPNN201 file is a working copy of the instructional features intermediate file. Its structure is identical to that of OG200. File CFPNN202 contains a working copy of the instructional feature data for this function or cue. File CFPNN203 contains the selected instructional features. Its structure is identical to that of OG203. File CFPNN204 contains the backwards links from instructional features to functions and cues. Its structure is identical to OG0C. The instructional features intermediate file (OG200.ITB), the instructional features data dictionary (OG201.ITB), the instructional features database (OG202.ITB), the instructional features data file

(OG203.ITB), and these working files are the only ones used in this routine.

OG20B2.IPF - review rule base variables and execute IF rule base for selected cues. This routine executes the instructional feature rule base for each cue selected previously. In addition to the files opened in OG20B1, the working cue file, CFPNN110 is used.

OG20B2A.IPF - create and update selected IF and intermediate files. This routine creates and/or updates the selected instructional feature file as well as the selected instructional feature intermediate file. It is intended to be executed only from OG20B2 as it requires files opened in that routine as well as variables established there.

OG20B2B.IPF - create and update selected IF backwards files. This routine creates and/or updates the backwards selected instructional feature intermediate file. It is intended to be executed only from OG20B2 as it requires files opened in that routine as well as variables established there.

OG20B3.IPF - set rule base variables and execute IF rule base for selected tasks. This routine, called by OG20B1 only, executes the instructional feature rule base for each function selected previously. In addition to the files opened in OG20B1, the working function file, CFPNN100 is used.

OG20B3A.IPF - create and update selected IF and intermediate files. This routine creates and/or updates the selected instructional feature file as well as the selected instructional feature intermediate file. It is intended to be executed only from OG20B3 as it requires files opened in that routine as well as variables established there.

OG20B3B.IPF - create and update selected IF backwards files. This routine creates and/or updates the backwards selected instructional feature intermediate file. It is intended to be executed only from OG20B3 as it requires files opened in that routine as well as variables established there.

OG20C1.IPF - answer rule base questions, instructional feature executive. This routine executes the instructional feature rule base setting all data input to the value "unknown." The routine uses work files named CFPNN200.ITB, CFPNN201, CFPNN202, CFPNN203, CFPNN204, as well as CFPNN100 (the selected functions database) and CFPNN110 (the selected cues database). The CFPNN200 file is an intermediate file to working copies of the instructional feature data. Its structure is identical to that of OG200. The CFPNN201 file is a working copy of the instructional features intermediate file. Its structure is identical to that of OG200. File CFPNN202 contains a working

copy of the instructional feature data for this function or cue. File CFPNN203 contains the selected instructional features. Its structure is identical to that of OG203. File CFPNN204 contains the backwards links from instructional features to functions and cues. Its structure is identical to OG0C. These working files are the only ones used in this routine.

OG20C2.IPF - set rule base variables and execute IF rule base for selected cues. This routine executes the instructional feature rule base for each cue selected previously. In addition to the files opened in OG20C1, the working cue file, CFPNN110 is used.

OG20C2A.IPF - create and update selected IF and intermediate files. This routine creates and/or updates the selected instructional feature file as well as the selected instructional feature intermediate file. It is intended to be executed only from OG20C2 as it requires files opened in that routine as well as variables established there.

OG20C2B.IPF - create and update selected IF backwards file. This routine creates and/or updates the backwards selected instructional feature intermediate file. It is intended to be executed only from OG20C2 as it requires files opened in that routine as well as variables established there.

OG20C3.IPF - set rule base variables and execute IF rule base for selected tasks/functions. This routine, called by OG20C1 only, executes the instructional feature rule base for each function selected previously. In addition to the files opened in OG20C1, the working function file, CFPNN100 is used.

OG20C3A.IPF - create and update selected IF and intermediate files. This routine creates and/or updates the selected instructional feature file as well as the selected instructional feature intermediate file. It is intended to be executed only from OG20C3 as it requires files opened in that routine as well as variables established there.

OG20C3B.IPF - create and update selected IF backwards file. This routine creates and/or updates the backwards selected instructional feature intermediate file. It is intended to be executed only from OG20C3 as it requires files opened in that routine as well as variables established there.

OG20D1.IPF - justify instructional feature results executive routine. This routine executes the instructional feature rule base by using the data input last for this "feeder" (cues or functions) and shows the justification for selecting the associated instructional features. The routine uses work files, named CFPNN200.ITB, CFPNN201, CFPNN202, CFPNN203 and CFPNN204, as well as CFPNN100 (the selected functions database) and CFPNN110

(the selected cues database). The CFPNN200 file is an intermediate file to working copies of the instructional feature data. Its structure is identical to that of OG200. The CFPNN201 file is a working copy of the instructional features intermediate file. Its structure is identical to that of OG200. CFPNN202 contains a working copy of the instructional feature data for this function or cue. CFPNN203 contains the selected instructional features. Its structure is identical to that of OG203. CFPNN204 contains the backwards links from instructional features to functions and cues. Its structure is identical to OG0C. These working files are the only ones used in this routine.

OG20D2.IPF - justify IF selections for cues. This routine executes the instructional feature rule base for the cue selected previously using the data provided previously. The justification for the instructional feature selections is thereby provided.

OG20D3.IPF - justify IF selections for functions. This routine executes the instructional feature rule base for the cue selected previously using the data provided previously. The justification for the instructional feature selections is thereby provided.

OG20E1.IPF - draw a graph showing results of IF rule base. This file draws a graph portraying the number of functions and cues that each selected instructional feature supports. The routine uses work files named CFPNN203.ITB and CFPNN204. File CFPNN203 contains the selected instructional features. Its structure is identical to that of OG203. File CFPNN204 contains the backwards links from instructional features to functions and cues. Its structure is identical to OG0C. These working files are the only ones used in this routine.

OG20F.IPF - instructional features maintenance menu. This routine presents the maintenance menu for the instructional features. The routine presents two options for selection: edit and add.

OG20F1.IPF - instructional feature edit routine. This routine edits records in the master instructional features data file (OG203.ITB). Users are prompted to select specific records to edit. The selected records are then presented in an instructional features data form, permitting the field variables to be changed. All changes made to the master file are permanent.

OG20F2.IPF - instructional feature add routine. This routine adds records to the master instructional features data file (OG203.ITB). An instructional features form is presented to the user so that new information can be entered for the field variables of the new record. Error checking routines are

included so that a newly added record does not conflict with an existing record in the file. The newly identified record is added to the master instructional features data file and user-specified links are added to the component intermediate file (OG220.ITB). All records created by this routine are permanently added to the master files.

OG21.IPF - display fidelity menu. This routine presents the fidelity menu.

OG21A1.IPF - automatic fidelity executive. This routine executes the fidelity rule base using the previous module's results and asking all appropriate questions necessary to arrive at adequate answers. The routine creates, if necessary, and uses work files named CFPNN210.ITB, CFPNN211, CFPNN212, CFPNN214 and CFPNN214. CFPNN210 is an intermediate file to working copies of the fidelity data. Its structure is identical to that of OG210. CFPNN211 is a working copy of the fidelity intermediate file. Its structure is identical to that of OG210. CFPNN212 contains a working copy of the fidelity data for this function or cue. CFPNN213 contains the selected fidelity dimensions. its structure is identical to that of OG213. CFPNN214 contains the backwards links from fidelities to functions and cues. Its structure is identical to OG0C. The fidelity intermediate file (OG210.ITB), the fidelity dictionary data file (OG211.ITB), the fidelity database (OG212.ITB), the fidelity dimension data file (OG213.ITB), and these working files are the only ones used in this routine.

OG21A2.IPF - execute fidelity rule base for selected cues. This routine executes the fidelity rule base for each cue selected previously. In addition to the files opened in OG21A1, the working cue file, CFPNN110 is used.

OG21A2A.IPF - create and update selected fidelity and intermediate files. This routine creates and/or updates the selected fidelity file as well as the selected fidelity intermediate file. It is intended to be executed only from OG21A2 as it requires files opened in that routine as well as variables established there.

OG21A2B.IPF - create and update selected fidelity backwards files. This routine creates and/or updates the backwards selected fidelity intermediate file. It is intended to be executed only from OG21A2 as it requires files opened in that routine as well as variables established there.

OG21A3.IPF - execute fidelity rule base for selected tasks. This routine executes the fidelity rule base for each function selected previously. In addition to the files opened in OG21A1, the working function file, CFPNN100 is used.

OG21A3A.IPF - create and update selected fidelity and intermediate files. This routine creates and/or updates the selected fidelity file as well as the selected fidelity intermediate file. It is intended to be executed only from OG21A3 as it requires files opened in that routine as well as variables established there.

OG21A3B.IPF - create and update selected fidelity backwards files. This routine creates and/or updates the backwards selected fidelity intermediate file. It is intended to be executed only from OG21A3 as it requires files opened in that routine as well as variables established there.

OG21A4.IPF - execute fidelity rule base for selected instructional features. This routine executes the fidelity rule base for each fidelity item selected previously. In addition to the files opened in OG21A1, the working function file, CFPNN100 is used.

OG21A4A.IPF - create and update selected fidelity and intermediate files. This routine creates and/or updates the selected fidelity file as well as the selected fidelity intermediate file. It is intended to be executed only from OG21A3 as it requires files opened in that routine as well as variables established there.

OG21A4B.IPF - create and update selected fidelity backwards files. This routine creates and/or updates the backwards selected fidelity intermediate file. It is intended to be executed only from OG21A3 as it requires files opened in that routine as well as variables established there.

OG21B1.IPF - execute the automatic with confirm fidelity executive routine. This routine executes the fidelity rule base by using the previous module's results and asking all appropriate questions necessary to arrive at adequate answers. The routine creates, if necessary, and uses work files, named CFPNN210.ITB, CFPNN211, CFPNN212, CFPNN214 and CFPNN214. The CFPNN210 file is an intermediate file to working copies of the fidelity data. Its structure is identical to that of OG210. The CFPNN211 file is a working copy of the fidelity intermediate file. Its structure is identical to that of OG210. CFPNN212 contains a working copy of the fidelity data for this function or cue. CFPNN213 contains the selected fidelities. Its structure is identical to that of OG213. CFPNN214 contains the backwards links from fidelities to functions and cues. Its structure is identical to OG0C. The fidelity intermediate file (OG210.ITB), the fidelity dictionary data file (OG211.ITB), the fidelity database (OG212.ITB), the fidelity dimension data file (OG213.ITB), and these working files are the only ones used in this routine.

OG21B2.IPF - execute fidelity rule base for selected cues.
This routine executes the fidelity rule base for each cue selected previously. In addition to the files opened in OG21B1, the working cue file, CFPNN110 is used.

OG21B2A.IPF - create and update selected fidelity and intermediate files. This routine creates and/or updates the selected fidelity file as well as the selected fidelity intermediate file. It is intended to be executed only from OG21B2 as it requires files opened in that routine as well as variables established there.

OG21B2B.IPF - create and update selected fidelity backwards files. This routine creates and/or updates the backwards selected fidelity intermediate file. It is intended to be executed only from OG21B2 as it requires files opened in that routine as well as variables established there.

OG21B3.IPF - execute fidelity rule base for selected tasks.
This routine executes the fidelity rule base for each function selected previously. In addition to the files opened in OG21B1, the working function file, CFPNN100 is used.

OG21B3A.IPF - create and update selected fidelity and intermediate files. This routine creates and/or updates the selected fidelity file as well as the selected fidelity intermediate file. It is intended to be executed only from OG21B3 as it requires files opened in that routine as well as variables established there.

OG21B3B.IPF - create and update selected fidelity backwards files. This routine creates and/or updates the backwards selected fidelity intermediate file. It is intended to be executed only from OG21B3 as it requires files opened in that routine as well as variables established there.

OG21B4.IPF - execute fidelity rule base for selected instructional features. This routine executes the fidelity rule base for each function selected previously. In addition to the files opened in OG21B1, the working function file, CFPNN100 is used.

OG21B4A.IPF - create and update selected fidelity and intermediate files. This routine creates and/or updates the selected fidelity file as well as the selected fidelity intermediate file. It is intended to be executed only from OG21B3 as it requires files opened in that routine as well as variables established there.

OG21B4B.IPF - create and update selected fidelity backwards files. This routine creates and/or updates the backwards selected fidelity intermediate file. It is intended to be

executed only from OG21B3 as it requires files opened in that routine as well as variables established there.

OG21C1.IPF - question mode fidelity executive. This routine executes the fidelity item rule base by ignoring the previous module's results and asking all appropriate questions necessary to arrive at adequate answers. The routine creates, if necessary, and uses work files, named CFPNN210.ITB, CFPNN211, CFPNN212, CFPNN213 and CFPNN214. The CFPNN210 file is an intermediate file to working copies of the fidelity data. Its structure is identical to that of OG210. The CFPNN211 file is a working copy of the fidelity items intermediate file. Its structure is identical to that of OG210. CFPNN212 contains a working copy of the fidelity data for this function or cue. CFPNN213 contains the selected fidelities. Its structure is identical to that of OG213. CFPNN214 contains the backwards links from fidelities to functions and cues. Its structure is identical to OG0C. The fidelity intermediate file (OG210.ITB), the fidelity dictionary data file (OG211.ITB), the fidelity database (OG212.ITB), the fidelity dimension data file (OG213.ITB), and these working files are the only ones used in this routine.

OG21C2.IPF - execute fidelity rule base for selected cues. This routine executes the fidelity rule base for each cue selected previously. In addition to the files opened in OG21C1, the working cue file, CFPNN110 is used.

OG21C2A.IPF - create and update selected fidelity intermediate files. This routine creates and/or updates the selected fidelity file as well as the selected fidelity intermediate file. It is intended to be executed only from OG21C2 as it requires files opened in that routine as well as variables established there.

OG21C2B.IPF - create and update selected fidelity backwards files. This routine creates and/or updates the backwards selected fidelity intermediate file. It is intended to be executed only from OG21C2 as it requires files opened in that routine as well as variables established there.

OG21C3.IPF - execute fidelity rule base for selected tasks. This routine executes the fidelity rule base for each function selected previously. In addition to the files opened in OG21C1, the working function file, CFPNN100 is used.

OG21C3A.IPF - create and update selected fidelity intermediate files. This routine creates and/or updates the selected fidelity file as well as the selected fidelity intermediate file. It is intended to be executed only from OG21C3 as it requires files opened in that routine as well as variables established there.

OG21C3B.IPF - create and update selected fidelity backwards files. This routine creates and/or updates the backwards selected fidelity intermediate file. It is intended to be executed only from OG21C3 as it requires files opened in that routine as well as variables established there.

OG21C4.IPF - execute fidelity rule base for selected IFs. This routine executes the fidelity rule base for each instructional feature identified previously. In addition to the files opened in OG21C1, the working function file, CFPNN100 is used.

OG21C4A.IPF - create and update selected fidelity and intermediate files. This routine creates and/or updates the selected fidelity file as well as the selected fidelity intermediate file. It is intended to be executed only from OG21C3 as it requires files opened in that routine as well as variables established there.

OG21C4B.IPF - create and update selected fidelity backwards files. This routine creates and/or updates the backwards selected fidelity intermediate file. It is intended to be executed only from OG21C3 as it requires files opened in that routine as well as variables established there.

OG21D1.IPF - justify fidelity results executive. This file, OG21D1.IPF executes the fidelity rule base by using the data input last for this "feeder" (cues or functions or instructional features) and shows the justification for selecting the associated fidelity dimensions. The routine uses work files, named CFPNN210.ITB, CFPNN211, CFPNN212, CFPNN213, CFPNN214, CFPNN100 (the selected functions database), and CFPNN110 (the selected cues database). The CFPNN210 file is an intermediate file to working copies of the fidelity data. Its structure is identical to that of OG210. The CFPNN211 file is a working copy of the fidelity intermediate file. Its structure is identical to that of OG210. CFPNN212 contains a working copy of the fidelity data for this function or cue. CFPNN213 contains the selected fidelities. Its structure is identical to that of OG213. CFPNN214 contains the backwards links from fidelities to functions and cues. Its structure is identical to OG0C. These working files are the only ones used in this routine.

OG21D2.IPF - justify fidelity selections for cues. This routine executes the fidelity rule base for the cue selected previously. The justification for the fidelity selections is thereby provided.

OG21D3.IPF - justify fidelity selections for functions. This routine executes the fidelity rule base for the function selected previously. The justification for the fidelity selections is thereby provided.

OG21D4.IPF - justify fidelity selections for IFs. This routine executes the fidelity rule base for the cue selected previously. The justification for the fidelity selections is thereby provided.

OG21E1.IPF - draws a graph showing results of fidelity rule base. This file draws a graph portraying the number of functions and cues and instructional features that each selected fidelity dimension supports. The routine uses work files, named CFPNN213.ITB, CFPNN214. CFPNN213 contains the selected fidelities. Its structure is identical to that of OG213. CFPNN214 contains the backwards links from fidelities to functions and cues. Its structure is identical to OG0C. These working files are the only ones used in this routine.

OG21F.IPF - fidelity dimensions maintenance menu. This routine presents the maintenance menu for the fidelity dimensions data. The routine presents two options for selection: edit and add.

OG21F1.IPF - fidelity dimensions edit routine. This routine edits records in the master fidelity dimensions data file (OG213.ITB). Users are prompted to select specific records to edit. The selected records are then presented in the appropriate fidelity dimension data form, permitting the field variables to be changed. All changes made to the master file are permanent.

OG21F2.IPF - fidelity dimensions add routine. This routine adds records to the master fidelity dimensions data file (OG213.ITB). A fidelity dimension form is presented to the user so that new information can be entered for the field variables of the new record. Error checking routines are included so that a newly added record does not conflict with an existing record in the file. The newly identified record is added to the master fidelity dimensions data file and user-specified links are added to the component intermediate file (OG220.ITB). All records created by this routine are permanently added to the master files.

OG22.IPF - display component selection menu. This routine presents the component selection menu.

OG22A1.IPF - automatic component selection executive. This routine performs the automatic component selection portion of the component selection module. This routine creates or uses the work file CFPNN220.ITB. It also creates or uses the backwards intermediate file CFPNN222. The working file contains duplicates of the desired records from the component database. Its structure is identical to that of OG221.

OG22A2.IPF - execute the component selection process from selected functions. This routine is called by OG22A1 and

processes the selected tasks to find the associated components. All files needed are opened in the executive routine. The only file used directly here is CFPNN100, the selected function file.

OG22A2A.IPF - update the backwards index pointing to functions. This routine creates and/or updates the backwards selected component intermediate file. It is intended to be execute only from OG22A1 as it requires files opened in that routine as well as variables established there.

OG22A3.IPF - execute the component selection process from selected cues. This routine is called by OG22A1 and processes the selected cues to find the associated components. All files needed are opened in the executive routine. The only file used directly here is CFPNN110, the selected cue file.

OG22A3A.IPF - update the backwards index pointing to cues. This routine creates and/or updates the backwards selected component intermediate file. It is intended to be executed only from OG22A1 as it requires files opened in that routine as well as variables established there.

OG22A4.IPF - execute the component selection process from selected IFs. This routine is called by OG22A1 and processes the selected tasks to find the associated components. Most files needed for this routine are opened in the executive routine. The only file opened directly in this routine is CFPNN203, the selected instructional feature file.

OG22A4A.IPF - update the backwards index pointing to IFs. This routine creates and/or updates the backwards selected component intermediate file. It is intended to be executed only from OG22A1 as it requires files opened in that routine as well as variables established there.

OG22A5.IPF - execute the component selection process from selected fidelity items. This routine is called by OG22A1 and processes the selected tasks to find the associated components. All files needed are opened in the executive routine. The only file used directly here is CFPNN213, the selected function file.

OG22A5A.IPF - update the backwards index pointing to fidelity items. This routine creates and/or updates the backwards selected component intermediate file. It is intended to be executed only from OG22A1 as it requires files opened in that routine as well as variables established there.

OG22B1.IPF - component selection. This routine presents a menu for the user selection of components. The list contains all components in the master database and allows the user to select (or deselect) the components for analysis. This routine creates, if necessary, and uses a work file, named CFPNN220.ITB. The

working file contains duplicates of the desired records from the component database. Its structure is identical to that of OG221.

OG22D1.IPF - component review. This routine presents the review component menu for the component selection module. This routine uses the work routine created by OG22A1.IPF, named CFPNN220.ITB. The working file contains duplicates of the desired records from the component database. Its structure is identical to that of OG221. The working file is the only one used in this routine.

OG22E1.IPF - component maintenance. This routine presents the maintenance menu for the component module by calling OG23C.IPF. That routine presents the maintenance menu for the master component database (OG221.ITB). The menu presents three options for selection: edit, add, and delete.

OG23.IPF - maintenance for components module. This routine presents the maintenance menus for the components module. It allows the user to select master files for editing. The users can select and modify any of the following master file databases: instructional features, fidelity dimensions, and components.

OG23A1.IPF - instructional feature edit routine. This routine edits records in the master instructional feature database (OG202.ITB). Users are prompted to select specific records to edit by selecting from the task/function list. The selected records are then presented in an instructional feature edit form, which permits the field variable of the record to be changed. All changes made to the master instructional feature database by this routine are permanent.

OG23B1.IPF - fidelity edit routine. This routine edits records in the master fidelity dimension database (OG212.ITB). Users are prompted through a multiple level routine in order to select specific records to edit. The first level identifies whether the fidelity information is associated with functions or cues. The second level selection lists either the function or cue records for selection. The third level selection occurs for each selected function or cue, in which the user is prompted for one or more fidelity dimension data sets for editing. The fidelity records associated with the selected functions or cues and fidelity dimensions are then presented in the appropriate fidelity edit form, which permits the field variable of the record to be changed. All changes made to the master fidelity dimension database by this routine are permanent.

OG23C.IPF - components maintenance menu. This routine presents the maintenance menu for the master component database (OG221.ITB). The menu presents three options for selection: edit, add, and delete.

OG23C1.IPF - components edit routine. This routine edits records in the master component database (OG221.ITB). Users are prompted to select specific records to edit. The selected records are then presented in an component edit form, which permits the field variable of the record to be changed. All changes made to the master component database by this routine are permanent.

OG23C2.IPF - components add routine. This routine adds records to the master component database (OG221.ITB). A component edit form is presented to the user so that field variables for the new record can be entered. Error checking routines are included so that a newly added record does not conflict with an existing record in the database. The newly identified record is added to the master component database. The user is then prompted to select individual functions (from OG104.ITB), cues (from OG111.ITB), instructional features (from OG203.ITB), and fidelity dimensions (from OG213.ITB) related to the new component. The component intermediate file (OG220.ITB) is updated with new records for each link selected. All records created by this routine are permanently added to the master component database.

OG23C3.IPF - component delete routine. This routine deletes selected records from the master component database (OG221.ITB). For each to-be-deleted component the link records in the component intermediate file (OG220.ITB) associating it to functions, cues, instructional features, and fidelity dimensions are altered to a "blank" value. All records deleted by this routine are permanently deleted from the master component database.

Module 3--Analyses

Program routines providing analysis capabilities are described below:

OG3.IPF - display analyses pull-down menu. This routine presents the component CERS menu.

OG3A.IPF - simulate analyses pull-down menu. This routine simulates the exercise of the analyses menu.

OG30.IPF - display component CERS menu. This routine presents the component CERS menu.

OG30A1.IPF - component cost comparison. This routine executes the component cost comparison spreadsheet. This routine uses the work routine CFPNN220.ITB.

OG30A2.IPF - draw graph comparing component cost. Draws component cost comparison graph.

OG30A3.IPF - activate component spreadsheet. This routine creates and displays a spreadsheet showing the component parameters for the selected family.

OG30B1.IPF - execute component effectiveness comparison. This routine executes the component effectiveness comparison spreadsheet portion of the component CERS module. This routine uses the work routine CFPNN220.ITB.

OG30B2.IPF - draw graph comparing component effectiveness. Draws component effectiveness comparison graph.

OG30B3.IPF - activate component spreadsheet. This routine creates and displays a spreadsheet showing the component parameters for the selected family.

OG30C1.IPF - execute component technology comparison. This routine executes the component technical risk comparison spreadsheet portion of the component CERS module. This routine uses the work routine CFPNN220.ITB.

OG30C2.IPF - draw graph comparing component technical risk. Draws component technical risk comparison graph.

OG30C3.IPF - activate component spreadsheet. This routine creates and displays a spreadsheet showing the component parameters for the selected family.

OG30D1.IPF - execute component schedule comparison. This routine executes the component schedule comparison spreadsheet portion of the component CERS module. This routine uses the work routine CFPNN220.ITB.

OG30D2.IPF - draw graph comparing component schedules. Draws component schedule comparison graph.

OG30D3.IPF - activate component spreadsheet. This routine creates and displays a spreadsheet showing the component parameters for the selected family.

OG30E1.IPF - execute overall component comparison. This routine executes the overall component comparison portion of the component CERS module. This routine uses the work routine CFPNN220.ITB.

OG30E2.IPF - draw graph showing overall component comparisons. Draws overall component comparison graph.

OG30E3.IPF - activate component spreadsheet. This routine creates and displays a spreadsheet showing the component parameters for the selected family.

OG31.IPF - display create systems menu. This routine presents the create systems menu.

OG31A1.IPF - automatic system creation. This routine performs the automatic system generation portion of the system module. This routine creates or uses the work routine CFPNN310.ITB, CFPNN311.ITB and CFPNN312.ITB. These work files and the components database, CFPNN220, are the only ones used in this routine.

OG31A2.IPF - create systems for "add" case. This routine creates systems from the working component file in the "add" case.

OG31A2A.IPF - create the forward system intermediate file. This routine creates and/or updates the system intermediate file. It is intended to be executed only from OG31A2 as it requires files opened in that routine as well as variables established there.

OG31A2B.IPF - create the backwards system intermediate file. This routine creates and/or updates the backwards system intermediate file. It is intended to be executed only from OG31A2 as it requires files opened in that routine as well as variables established there.

OG31A3.IPF - create systems for "start over" case. This routine creates systems from the working component file in the "starting over" case.

OG31A3A.IPF - create the forward system intermediate file. This routine creates and/or updates the system intermediate file. It is intended to be executed only from OG31A3 as it requires files opened in that routine as well as variables established there.

OG31A3B.IPF - create the backwards system intermediate file. This routine creates and/or updates the backwards system intermediate file. It is intended to be executed only from OG31A3 as it requires files opened in that routine as well as variables established there.

OG31A4.IPF - allow naming of generated systems. This routine presents the generated systems and allows any of them to be named.

OG31A5.IPF - allow deselecting generated systems. This routine presents the generated systems and allows any of them to be deselected.

OG31A6.IPF - calculate system parameters. This routine calculates the system parameters on the basis of the components

contained in the system. There is a known problem in this routine that is based on calculating effectiveness for components that do not have any task/functions associated with them. The effectiveness is derived from the associated task/functions, and if none are present the effectiveness is set to zero (0), leading to an absurd result. The obvious solution is for the user not to include any components that have no reason for inclusion, that is components that are not driven to inclusion by a relationship with task/functions.

OG31B1.IPF - select components for system generation. This routine performs the select system components portion of the system module. This routine creates or uses the work routine CFPNN310.ITB, CFPNN311.ITB and CFPNN312.ITB. These work files and the components database, CFPNN220, are the only ones used in this routine.

OG31B2.IPF - look up systems for "add" case. This routine creates systems from the working component file in the "add" case.

OG31B2A.IPF - create the forward system intermediate file. This routine creates and/or updates the system intermediate file. It is intended to be executed only from OG31B2 as it requires files opened in that routine as well as variables established there.

OG31B2B.IPF - create the backwards system intermediate file. This routine creates and/or updates the backwards system intermediate file. It is intended to be executed only from OG31B2 as it requires files opened in that routine as well as variables established there.

OG31B3.IPF - look up systems for "start over" case. This routine creates systems from the working component file in the "starting over" case.

OG31B3A.IPF - create the forward system intermediate file. This routine creates and/or updates the system intermediate file. It is intended to be executed only from OG31B3 as it requires files opened in that routine as well as variables established there.

OG31B3B.IPF - create the backward system intermediate file. This routine creates and/or updates the backwards system intermediate file. It is intended to be executed only from OG31B3 as it requires files opened in that routine as well as variables established there.

OG31B4.IPF - allow naming of generated systems. This routine presents the generated systems and allows any of them to be named.

OG31B5.IPF - allow deselecting generated systems. This routine presents the generated systems and allows any of them to be deselected.

OG31B6.IPF - calculate system parameters. This routine calculates the system parameters on the basis of the components contained in the system. (See OG31A6).

OG31D1.IPF - review systems. This routine presents the review systems menu for the system generation module. This routine uses the generated systems working file CFPNN311.

OG32.IPF - display system CERS menu. This routine presents the system CERS menu.

OG32A1.IPF - system cost comparison. This routine executes the system cost comparison spreadsheet portion of the system CERS module. This routine uses the work routine CFPNN311.ITB.

OG32A2.IPF - draw graph comparing system cost. Draws system cost comparison graph.

OG32A3.IPF - activate system spreadsheet. This routine creates and displays a spreadsheet showing the system parameters.

OG32B1.IPF - system effectiveness comparison. This routine executes the system effectiveness comparison spreadsheet portion of the system CERS module.

OG32B2.IPF - draw graph comparing system effectiveness. Draws component effectiveness comparison graph.

OG32B3.IPF - activate system spreadsheet. This routine creates and displays a spreadsheet showing the system parameters.

OG32C1.IPF - system technological comparison. This routine executes the system technical risk comparison portion of the system CERS module.

OG32C2.IPF - draw graph comparing system technical risk. This routine draws the graph showing the technical risk of each system.

OG32C3.IPF - activate system spreadsheet. This routine creates and displays a spreadsheet showing the system parameters.

OG32D1.IPF - system schedule comparison. This routine executes the system schedule comparison portion of the system CERS module. This routine uses the work routine CFPNN311.ITB.

OG32D2.IPF - draw graph comparing system schedules. Draws system schedule comparison graph.

OG32D3.IPF - activate system spreadsheet. This routine creates and displays a spreadsheet showing the system parameters.

OG32E1.IPF - overall system comparison. This routine executes the overall system comparison portion of the system CERS module. This routine uses the work routine CFPNN311.ITB.

OG32E2.IPF - draw graph showing overall system comparisons. Draws overall system comparison graph.

OG32E3.IPF - activate system spreadsheet. This routine creates and displays a spreadsheet showing the system parameters.

OG32F1.IPF - execute system comparison. This routine executes the life cycle cost comparison portion of the system CERS module.

OG32F2.IPF - graph system comparison. Draws overall system comparison graph.

OG32F3.IPF - activate system spreadsheet. This routine creates and displays a spreadsheet showing the system parameters.

OG33.IPF - system maintenance menu. This routine displays a message informing the user that the edit routines for systems have not been developed or implemented.

Module 4--Documentation

Program routines providing documentation assistance are described below:

OG4.IPF - display documentation pull-down menu. This routine presents the documentation menu.

OG40.IPF - generate outline. This routine, calls OG40.IPF in order to generate a completely new outline for all the modules thus far run. Before the outline is generated the working outline file, CFPNN400, and its index is compressed to delete any previously generated outline entries. At this point the only entries remaining in the working outline file are the master entries.

OG400.IPF - make outline entries (function module). This routine causes outline entries to be made in the project's outline file, CFPNN400, and its index file. The particular entries are those that pertain to the function selection module.

OG401.IPF - make outline entries (cue module). This routine causes outline entries to be made in the project's outline file, CFPNN400, and its index file. The particular entries are those that pertain to the cue selection module.

OG402.IPF - make outline entries (simulation checklist). This routine causes outline entries to be made in the project's outline file, CFPNN400, and its index file. The particular entries are those that pertain to the simulation checklist module.

OG403.IPF - make outline entries (IF module). This routine causes outline entries to be made in the project's outline file, CFPNN400, and its index file. The particular entries are those that pertain to the instructional features module.

OG404.IPF - make outline entries (fidelity module). This routine causes outline entries to be made in the project's outline file, CFPNN400, and its index file. The particular entries are those that pertain to the instructional features module.

OG405.IPF - make outline entries (component module). This routine causes outline entries to be made in the project's outline file, CFPNN400, and its index file. The particular entries are those that pertain to the component selection module.

OG406.IPF - make outline entries (component analysis). This routine causes outline entries to be made in the project's outline file, CFPNN400, and its index file. The particular entries are those that pertain to the component analysis module.

OG407.IPF - make outline entries (create systems). This routine causes outline entries to be made in the project's outline file, CFPNN400, and its index file. The particular entries are those that pertain to the cue selection module.

OG408.IPF - make outline entries (system analysis). This routine causes outline entries to be made in the project's outline file, CFPNN400, and its index file. The particular entries are those that pertain to the systems analysis module.

OG42.IPF - edit outline template. This routine allows perusal of the outline template, editing its entries, and adding additional outline items

OG43.IPF - print outline function. This routine prints out an outline for the current project in the outline module. This routine uses the file CFPNN400, the working copy of the outline and its index.

OG44.IPF - print outline to text file. This routine, OG44.IPF, prints an outline to the file OUTLINE.PRN for the current project. The ASCII file is written to the project directory.

INDEX

add 15, 17-18, 19, 25, 30,
 33, A-1, A-2, A-3, A-4,
 A-6, A-7
 analyses 5, 10, 33
 analysis 39, 40, A-4, A-6,
 B-2
 ASI 1, 14, 17, 18, A-1, A-
 2, A-6, B-3
 CERS 1, 5, 11, 34, 35, 37,
 38, A-4, A-5
 checklist 11, 13, 16, 39,
 A-1, A-6, B-2, B-4, B-11
 component 1, 4, 5, 11, 13,
 14, 19, 20, 25, 30-37, 39,
 A-1, A-4, A-5, A-6, A-7,
 B-1, B-2, B-4, B-6, B-8,
 B-9, B-10, B-11, C-2, C-4
 concept formulation 1, 2,
 cost 1, 2, 34, 37, 38, A-4,
 A-5, B-2, B-9, B-10
 delete 4, 11, 13, 15-20,
 32, 33, 39, A-1, A-2,
 A-4
 device 1, 2, 4, A-2, A-4,
 B-3, B-10
 directory 2, 3, 6, 9, 10,
 40, B-2
 documentation 1, 5, 38, A-6
 edit 15-19, 24, 25, 30, 32,
 33, 38, 40, A-1, A-2, A-
 3, A-4, A-6, A-7
 effectiveness 1, 5, 34, 36,
 37, A-5, B-2, B-3
 fidelity 1, 5, 10, 11, 13,
 14, 19, 20, 25-33, 39,
 A-1, A-3, A-4, A-6, A-7,
 B-1, B-2, B-4, B-5, B-6,
 B-7, B-8, B-10, B-11,
 C-1, C-2, C-3, C-4
 generate(s) 5, 39, A-6, C-3
 generation 35, 36, 37, A-5
 graph 24, 30, 31, 34, 35,
 37, 38, A-3, A-4, A-5,
 A-6
 GURU 1-3, 5, 6, 11, B-1, C-
 1, C-2,
 IF 13, 21, 22-24, 28, 30,
 31, 39, A-2, A-3, A-
 4, A-6, A-7, B-1, B-2,
 B-3, B-4, B-5, B-6, B-
 7, B-11
 index 9, 10-14, 16, 31, 32,
 39, 40, A-1, A-4, B-1,
 B-2, B-11
 installation 2, 3, 6
 instructional feature(s) 1,
 5, 10, 11, 13, 14,
 19-26, 27, 29-33, 39, A-
 1, A-2, A-3, A-4, A-6,
 B-5, B-6, B-11, C-1,
 C-2, C-3, C-4
 intermediate file 14, 15,
 17-33, 35-37, A-2, A-3,
 A-4, A-5, A-6, B-1, B-2,
 B-3, B-4, B-5, B-6, B-7,
 B-8, B-9, B-10, B-11
 link(s) 1, 10, 11, 14, 16-20,
 22-25, 27-30, 33, A-6, A-
 7, B-1, B-3, B-4, B-5, B-
 6, B-7, B-8, B-9, B-10, B-
 11, B-12, C-1, C-3, C-4
 maintenance 15-20, 24, 30,
 32, 33, 38, A-1, A-2,
 A-3, A-4, A-6, A-7
 master data file 9
 master file 17, 25, 30, 32
 MOS 1, 9, 14, 17, A-1, A-2,
 B-3
 outline 5, 10, 11, 12, 14,
 39, 40, A-1, A-6, B-10,
 B-12
 pointers 9, 10
 print 4, 5, 13, 14, 40,
 A-1, A-6
 risk 1, 5, 34, 38, A-5,
 B-2, B-9, B-10
 rule base 4, 9, 13, 20-30,
 A-2, A-3, A-4, A-6, A-7,
 B-5, B-6, B-7, C-1, C-2,
 C-3, C-4
 schedule 1, 5, 34, 38, A-5,
 B-2, B-9, B-10
 spreadsheet 1, 34, 35, 37,
 38, A-5, A-6
 SQI 1, 14, 15, 17-19, A-1,
 A-2, A-6, B-3, B-4
 subdirectory(ies) 3, 9, 10
 technical risk 5, 34, 38,

A-5, B-2, B-9, B-10
working file 5, 9, 10, B-
11, B-12

References

- Elder, B. L., Sticha, P. J., Page, G. L., & Singer, M. J. (In Preparation). Concept Formulation Process Aid for Trade-Off Determination: User's Guide (ARI Research Product 93-06). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.
- Micro Data Base Systems, Inc. (July 1991). GURU Information Management. [Computer Program], Lafayette, IN.
- Micro Data Base Systems, Inc. (July 1991). GURU Expert Systems. [Computer Program], Lafayette, IN.
- Sticha, P. J., Gibbons, S., & Singer, M. J. (In Preparation). Development of a Concept Formulation Process Aid for Analyzing Training Requirements and Developing Weapon Systems Training Devices. (ARI Research Report 1637). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

APPENDIX A

CFP-Aid Interpreter Program Files

Program Files Described in Body of Report

cfp-aid.ipf	the main driver routine, show main menu
oga.ipf	simulate the main menu
og0.ipf	display the file pull-down menu
og0a.ipf	simulate the file pull-down menu
og001.ipf	create new project
og01.ipf	display open menu
og01a1.ipf	project menu
og01b1.ipf	select project description
og021.ipf	save as...
og031.ipf	project status
og041.ipf	delete project
og05.ipf	display print menu
og05a1.ipf	print project list
og05b1.ipf	print task/function database
og05c1.ipf	print cue database
og05d1.ipf	print simulation checklist
og05e1.ipf	print instructional features data for functions
og05e2.ipf	print instructional features data for cues
og05f1.ipf	print fidelity data for functions
og05f2.ipf	print fidelity data for cues
og05f3.ipf	print fidelity data for instructional features
og05g1.ipf	print component database
og05h1.ipf	print outline template
og05i1.ipf	print all
og1.ipf	display function pull-down menu
og1a.ipf	simulate function pull-down menu
og10.ipf	display function determination menu
og10a1.ipf	MOS selection
og10a2.ipf	ASI selection
og10a3.ipf	SQI selection
og10a4.ipf	review taxonomy choices
og10b1.ipf	task/function selection
og10d1.ipf	review tasks/functions
og10e1.ipf	working task/function maintenance menu
og10e2.ipf	working task/function edit routine
og10e3.ipf	working task/function add message
og10e4.ipf	working task/function delete message
og11.ipf	display the cue source menu
og11a1.ipf	automatic cue source
og11a2.ipf	update cue to function index
og11b1.ipf	cue selection
og11d1.ipf	cue review
og11e.ipf	cue maintenance menu - branch to og13e.ipf
og121.ipf	simulation checklist

og13.ipf	maintenance menu for requirement module
og13a.ipf	MOS maintenance menu
og13a1.ipf	MOS edit routine
og13a2.ipf	MOS add routine
og13a3.ipf	MOS delete routine
og13b.ipf	ASI maintenance menu
og13b1.ipf	ASI edit routine
og13b2.ipf	ASI add routine
og13b3.ipf	ASI delete routine
og13c.ipf	SQI maintenance menu
og13c1.ipf	SQI edit routine
og13c2.ipf	SQI add routine
og13c3.ipf	SQI delete routine
og13d.ipf	task/function maintenance menu
og13d1.ipf	task/function edit routine
og13d2.ipf	task/function add routine
og13d3.ipf	task/function delete routine
og13e.ipf	cue maintenance menu
og13e1.ipf	cue edit routine
og13e2.ipf	cue add routine
og13e3.ipf	cue delete routine
og2.ipf	display the device pull-down menu
og2a.ipf	simulate the device pull-down menu
og20.ipf	display the instructional feature menu
og20a1.ipf	automatic instructional feature executive
og20a2.ipf	set rule base variables and execute IF rule base for selected cues
og20a2a.ipf	create and update selected IF and intermediate files
og20a2b.ipf	create and update selected IF backwards files
og20a3.ipf	set rule base variables and execute IF rule base for selected tasks
og20a3a.ipf	create and update selected IF and intermediate files
og20a3b.ipf	create and update selected IF backwards files
og20b1.ipf	automatic, with confirmation, instructional feature executive
og20b2.ipf	set rule base variables and execute IF rule base for selected cues
og20b2a.ipf	create and update selected IF and intermediate files
og20b2b.ipf	create and update selected IF backwards files
og20b3.ipf	set rule base variables and execute IF rule base for selected tasks
og20b3a.ipf	create and update selected IF and intermediate files
og20b3b.ipf	create and update selected IF backwards files
og20c1.ipf	answer rule base questions, instructional feature executive
og20c2.ipf	set rule base variables and execute IF rule base for selected cues

og20c2a.ipf	create and update selected IF and intermediate files
og20c2b.ipf	create and update selected IF backwards files
og20c3.ipf	set rule base variables and execute IF rule base for selected tasks
og20c3a.ipf	create and update selected IF and intermediate files
og20c3b.ipf	create and update selected IF backwards files
og20d1.ipf	justify instructional feature results executive
og20d2.ipf	justify IF selections for cues
og20d3.ipf	justify IF selections for functions
og20e1.ipf	draw a graph showing results of IF rule base
og20f.ipf	instructional feature data maintenance menu
og20f1.ipf	instructional features edit routine
og20f2.ipf	instructional features add routine
og21.ipf	display the fidelity menu
og21a1.ipf	automatic fidelity executive
og21a2.ipf	run fidelity rule base for selected cues
og21a2a.ipf	create and update selected fidelity and intermediate files
og21a2b.ipf	create and update selected fidelity backwards files
og21a3.ipf	run fidelity rule base for selected tasks
og21a3a.ipf	create and update selected fidelity and intermediate files
og21a3b.ipf	create and update selected fidelity backwards files
og21a4.ipf	run fidelity rule base for selected IFs
og21a4a.ipf	create and update selected fidelity and intermediate files
og21a4b.ipf	create and update selected fidelity backwards files
og21b1.ipf	automatic, with confirmation, fidelity executive
og21b2.ipf	run fidelity rule base for selected cues
og21b2a.ipf	create and update selected fidelity and intermediate files
og21b2b.ipf	create and update selected fidelity backwards files
og21b3.ipf	run fidelity rule base for selected tasks
og21b3a.ipf	create and update selected fidelity and intermediate files
og21b3b.ipf	create and update selected fidelity backwards files
og21b4.ipf	run fidelity rule base for selected IFs
og21b4a.ipf	create and update selected fidelity and intermediate files
og21b4b.ipf	create and update selected fidelity backwards files
og21c1.ipf	executes the fidelity executive routine
og21c2.ipf	run fidelity rule base for selected cues
og21c2a.ipf	create and update selected fidelity and intermediate files

og21c2b.ipf	create and update selected fidelity backwards files
og21c3.ipf	run fidelity rule base for selected tasks
og21c3a.ipf	create and update selected fidelity and intermediate files
og21c3b.ipf	create and update selected fidelity backwards files
og21c4.ipf	run fidelity rule base for selected IFs
og21c4a.ipf	create and update selected fidelity and intermediate files
og21c4b.ipf	create and update selected fidelity backwards files
og21d1.ipf	justify fidelity results executive routine
og21d2.ipf	justify fidelity selections for cues
og21d3.ipf	justify fidelity selections for functions
og21d4.ipf	justify fidelity selections for IFs
og21e1.ipf	draw a graph showing results of fidelity rule base
og21f.ipf	fidelity dimension maintenance menu
og21f1.ipf	fidelity dimension edit routine
og21f2.ipf	fidelity dimension add routine
og22.ipf	display the component selection menu
og22a1.ipf	automatic component selection
og22a2.ipf	run the component selection process from selected functions
og22a2a.ipf	update the backwards index pointing to functions
og22a3.ipf	run the component selection process from selected cues
og22a3a.ipf	update the backwards index pointing to cues
og22a4.ipf	run the component selection process from selected IFs
og22a4a.ipf	update the backwards index pointing to IFs
og22a5.ipf	run the component selection process from selected fidelity items
og22a5a.ipf	update the backwards index pointing to fidelity items
og22b1.ipf	run the component selection routine
og22d1.ipf	run the component review routine
og22e1.ipf	run the component maintenance routine
og23.ipf	data maintenance menu for device module
og23a1.ipf	instructional feature data edit routine
og23b1.ipf	fidelity data edit routine
og23c.ipf	component data maintenance menu
og23c1.ipf	component data edit routine
og23c2.ipf	component data add routine
og23c3.ipf	component data delete routine
og3.ipf	display the analysis pull-down menu
og3a.ipf	simulate the analysis pull-down menu
og30.ipf	display the component CERS menu
og30a1.ipf	run the component cost comparison routine
og30a2.ipf	draw graph comparing component cost

og30a3.ipf	show a spreadsheet of component data
og30b1.ipf	run the component effectiveness comparison routine
og30b2.ipf	draw graph comparing component effectiveness
og30b3.ipf	show a spreadsheet of component data
og30c1.ipf	run the component technical risk comparison routine
og30c2.ipf	draw graph comparing component technical risk
og30c3.ipf	show a spreadsheet of component data
og30d1.ipf	run the component schedule comparison routine
og30d2.ipf	draw graph comparing component schedules
og30d3.ipf	show a spreadsheet of component data
og30e1.ipf	run the overall component comparison routine
og30e2.ipf	draw graph showing overall component comparisons
og30e3.ipf	create and displays a spreadsheet of overall comparison
og31.ipf	display the create systems menu
og31a1.ipf	run the automatic system creation routine
og31a2.ipf	create systems for "add" case
og31a2a.ipf	create the forward system intermediate file
og31a2b.ipf	create the backwards system intermediate file
og31a3.ipf	create systems for "start over" case
og31a3a.ipf	create the forward system intermediate file
og31a3b.ipf	create the backwards system intermediate file
og31a4.ipf	allow naming of generated systems
og31a5.ipf	allow deselecting generated systems
og31a6.ipf	calculate system parameters from components
og31b1.ipf	run the selection system generation routine
og31b2.ipf	look up systems for "add" case
og31b2a.ipf	create the forward system intermediate file
og31b2b.ipf	create the backwards system intermediate file
og31b3.ipf	look up systems for "start over" case
og31b3a.ipf	create the forward system intermediate file
og31b3b.ipf	create the backward system intermediate file
og31b4.ipf	allow naming of generated systems
og31b5.ipf	allow deselecting generated systems
og31b6.ipf	calculate system parameters from components
og31d1.ipf	run the review systems routine
og32.ipf	display the system CERS menu
og32a1.ipf	run the system cost comparison routine
og32a2.ipf	draw graph comparing system cost
og32a3.ipf	provide a spreadsheet of system parameters
og32b1.ipf	run the system effectiveness comparison routine
og32b2.ipf	draw graph comparing system effectiveness
og32b3.ipf	provide a spreadsheet of system parameters
og32c1.ipf	run the system technical risk comparison routine
og32c2.ipf	draw graph comparing system technical risk
og32c3.ipf	provide a spreadsheet of system parameters
og32d1.ipf	run the system schedule comparison routine
og32d2.ipf	draw graph comparing system schedules
og32d3.ipf	provide a spreadsheet of system parameters
og32e1.ipf	run the overall system comparison routine

og32e2.ipf	draw graph showing overall system comparisons
og32e3.ipf	provide a spreadsheet of system parameters
og32f1.ipf	run the integrated system comparison routine
og32f2.ipf	draw a graph showing integrated system comparisons
og32f3.ipf	provide a spreadsheet of system parameters
og33.ipf	system maintenance menu - no options implemented
og4.ipf	display the documentation pull-down menu
og40.ipf	generate outline
og400.ipf	make outline entries (function module)
og401.ipf	make outline entries (cue module)
og402.ipf	make outline entries (simulation checklist)
og403.ipf	make outline entries (IF module)
og404.ipf	make outline entries (fidelity module)
og405.ipf	make outline entries (component module)
og406.ipf	make outline entries (component analysis)
og407.ipf	make outline entries (create systems)
og408.ipf	make outline entries (system analysis)
og42.ipf	edit outline template routine
og43.ipf	print outline function
og44.ipf	print outline to text file

Additional Program Files

og10e1.ipf	performs the function determination maintenance routine
og10e1a.ipf	gets a task description and associated data
og10e1a1.ipf	gets an ASI for task maintenance
og10e1a2.ipf	gets an SQI for task maintenance
og10e1a3.ipf	does task maintenance for other modules
og10e1b.ipf	edits a task description and associated data
og10e1b1.ipf	updates links to cues for a task
og10e1b2.ipf	updates links to IF data for a task
og10e1b3.ipf	updates links to fidelity data for a task
og10e1b4.ipf	updates links to components for a task
og11e1.ipf	cue maintenance
og11e1a.ipf	get cue description and associated data
og11e1a1.ipf	do cue maintenance for other modules
og11e1b.ipf	edit cue description and associated data
og11e1b1.ipf	update links to IF data for a cue
og11e1b2.ipf	update links to fidelity data for a cue
og11e1b3.ipf	update links to components for a cue
og131.ipf	maintenance routine for tasks/functions
og20f1.ipf	instructional feature maintenance
og20f2.ipf	allow selection of add or edit for IF maintenance
og20f2a.ipf	get a new set of IF rule base data
og20f2a1.ipf	get a new set of IF rule base data for other modules
og20f2a2.ipf	permit entry of new rule base name and new data dictionary
og20f3.ipf	allow selection of add or edit for IF items

og20f3a.ipf	get IF item information
og20f3a1.ipf	get new IF item information for other modules
og20f3b.ipf	edit IF item information
og20f3b1.ipf	update links to fidelity data for an IF
og20f3b2.ipf	update links to components for an IF
og21f1.ipf	fidelity maintenance
og21f2.ipf	allow selection of add or edit for fidelity maintenance
og21f2a.ipf	get a new set of fidelity rule base data
og21f2a1.ipf	get a new set of fidelity rule base data for other modules
og21f2a2.ipf	permit entry of new rule base name and new data dictionary
og21f3.ipf	allow selection of add or edit for fidelity items
og21f3a.ipf	get fidelity item information
og21f3a1.ipf	get new fidelity item information for other modules
og21f3b.ipf	edit fidelity item information
og21f3b1.ipf	update links to components for a fidelity dimension
og22e1.ipf	performs the component maintenance routine
og22e1a.ipf	gets new component information
og22e1a1.ipf	allows entry of new component information from other modules
og22e1b.ipf	updates links to components from fidelity dimensions

APPENDIX B

CFP-Aid Database File Structures and Index File Keys

The database fields, types, sizes, and a description of the information provided by the field, or a description of how the field information is used, is provided in the following listings. The aliases provide the naming convention used in the program files for referring to the files. The index file is identified by extension (.ind) and the key for the database index file is provided at the end of each listing. This index field is particularly important to rapid access of individual records in each database. The CFP-Aid may produce errors if the index files have been corrupted. In that case, consult the GURU manuals for instructions on re-indexing files.

File: OGOA Memory file - alias memory

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
nextsn	char	2	next available project serial number
inits	char	3	project identifier ("CFP")
syscolor	char	8	system colors (e.g., like in menu function)
link1	char	6	next available number for link 1 -- to function intermediate file
link2	char	6	next available number for link 2 -- function intermediate file to functions
link3	char	6	next available number for link 3 -- functions to cue intermediate file
link4	char	6	next available number for link 4 -- cue intermediate file to cues
link5	char	6	next available number for link 5 -- functions to IF intermediate file
link6	char	6	next available number for link 6 -- IF intermediate file to IF data file
link7	char	6	next available number for link 7 -- functions to fidelity intermediate file
link8	char	6	next available number for link 8 -- fidelity intermediate file to fidelity data
link9	char	6	next available number for link 9 -- functions to component intermediate file
link10	char	6	next available number for link 10 -- component intermediate file to comps
link11	char	6	next available number for link 11 -- cues to IF intermediate file
link12	char	6	next available number for link 12 -- functions/cues to working IF intermediate file
link13	char	6	next available number for link 13 -- cue to fidelity intermediate file
link14	char	6	next available number for link 14 -- IF working intermediate file to selected IFs
link15	char	6	next available number for link 15 -- cues

link16	char	6	to component intermediate file next available number for link 16 -- link to working IF data
link17	char	6	next available number for link 17 -- IF to fidelity intermediate file
link18	char	6	next available number for link 18 -- functions/cues/IFs to working fidelity intermediate file
link19	char	6	next available number for link 19 -- IF to component intermediate file
link20	char	6	next available number for link 20 -- fidelity working intermediate file to selected fids
link21	char	6	next available number for link 21 -- fidelity to component intermediate file
link22	char	6	next available number for link 22 -- link to working fidelity data
link23	char	6	next available number for link 23 -- comps to system intermediate file
link24	char	6	next available number for link 24 -- system intermediate file to system
link25	char	6	next available number for link 25 -- tasks to checklist items

File: OG0B Project file alias project

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
title	char	30	project title
projdesc	char	60	project description
datetime	char	16	today() + time()
projsn	char	2	project serial number
nograds	num	8	number of required trainees per year
projdir	char	64	project directory
flags	char	30	flags showing which modules have been run
cwt	num	8	cost analysis weight
ewt	num	8	effectiveness analysis weight
rwt	num	8	technical risk analysis weight
swt	num	8	schedule analysis weight
lifecycl	num	8	planning lifecycle
stdadjrt	num	8	standard adjustment rate
index file OG0B, key is title			

File: OG0C structure for backwards link files

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
source	char	2	source module code
blinkin	char	6	incoming link (current in source)
intblink	char	6	internal link to next in chain (next avail. in source)
blinkout	char	6	link out to target (current in target)

File: OGOB learning model memory file (one record only, no index)

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
stdadj	num	8	standard adjustment rate
lmexp	num	8	learning model exponent
maxifeff	num	8	maximum IF effectiveness
maxnoif	num	8	maximum number of IFs
setupsav	num	8	setup savings
ndevutil	num	8	new device utilization
ndevlc	num	8	new device lifecycle

File: OG100 MOS database alias MOS

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
mos	char	4	MOS number
branch	char	1	Branch code (E - enlisted, O - officer, W - warrant officer)
mosdesc	char	40	Description of MOS
index file OG100, key is branch + MOS			

File: OG101 ASI database alias ASI

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
mos	char	4	Related MOS number
asi	char	2	Additional Skill Identifier number
branch	char	1	Branch code
asidesc	char	40	Description of Additional Skill
link1	char	6	link to functions intermediate file (links are right justified numerals in strings)
index file OG101, key is branch + MOS + ASI			

File: OG102 SQI database alias SQI

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
sqi	char	2	Skill Qualification Identifier number
branch	char	1	Branch code
sqidesc	char	40	Description of Skill Qualification
link1	char	6	link to functions intermediate file (links are right justified numerals in strings)
index file OG102, key is branch + SQI			

File: OG103 functions intermediate file

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link1	char	6	incoming link from ASI and SQI databases
intlink	char	6	internal link to next in chain
link2	char	6	outgoing link to functions database
index file OG103, key is link1			

File OG104 Functions file alias FUNCT

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link2	char	6	incoming link from ASI, SQI and backwards link
fnDESC	char	40	function or task description
link3	char	6	link to cue intermediate file
link5	char	6	link to IF intermediate file
link7	char	6	link to fidelity intermediate file
link9	char	6	link to component intermediate file
link12	char	6	link to working IF intermediate file
link18	char	6	link to working fidelity intermediate file
entry	num	8	entry level of training
std	num	8	training standard
c_hours	num	8	classroom hours with no simulator
eq_hrs	num	8	equipment hours used in training without simulator
set_hrs	num	8	equipment setup hours with no simulator
lm_exp	num	8	learning model exponent for this task
taskcrit	num	8	task criticality

index file OG104, key is link2

File OG110 cue intermediate file alias inter

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link3	char	6	incoming link from functions databases
intlink	char	6	internal link to next in chain
link4	char	6	outgoing link to cue database

index OG110, key is link3

File OG111 cue file, alias cue

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link4	char	6	incoming link from functions
cuedesc	char	40	description of this cue
cuecrit	num	8	criticality of this cue
link11	char	6	link to IF intermediate file
link13	char	6	link to fidelity intermediate file
link12	char	6	link to working IF intermediate file
link15	char	6	link to component intermediate file
link18	char	6	link to fidelity working intermediate file

index file OG111, key is link4

File OG120 simulation checklist file, alias cklist

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
question	char	76	this contains text of question
recno	num	8	record number of this record
link2	char	6	identifier for each function
priority	char	1	this contains the priority of the question A highest
rating1	num	8	rating for one in group
rating2	num	8	rating for two in group
rating3	num	8	rating for three in group
rating4	num	8	rating for four in group
rating5	num	8	rating for five in group
index file OG120, key is link2 + priority			

File OG200 instructional feature intermediate file

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
source	str	2	source module code
linkin	str	6	incoming link from functions and cue databases
intlink	str	6	internal link to next in chain
link6	str	6	outgoing link to IF database
index file OG200, key is source + linkin			

File OG201 data dictionary for variable name versus field names

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
rulebase	char	8	name of rule base
fldname	char	8	name of field
varname	char	8	corresponding name of variable
vardesc	char	50	gives description of parameter for screen preview
vartype	char	70	gives menu selections for screen preview
index file OG201, key is rulebase			

File OG202 instructional feature database, alias if

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link6	str	6	incoming link from IF intermediate file
rulebase	str	8	name of rule base to use for this item
if_var1	num	8	stage of learning
if_var2	num	8	performance standard
if_var3	num	8	continuous movement activity
if_var4	num	8	procedures activity
if_var5	num	8	perception activity
if_var6	num	8	decision making/rule using activities
if_var7	num	8	using symbolic information activities
if_var8	num	8	voice communications activities
if_var9	num	8	discrete behavior detectable by computer
if_var10	num	8	intrinsic feedback
if_var11	num	8	probability of crash or kill
if_var12	num	8	cue saliency
if_var13	num	8	situational awareness
if_var14	num	8	number of steps in task
if_var15	num	8	time sharing -- low, medium, high
if_var16	num	8	instructor's role
if_var17	num	8	performance tolerances
if_var18	num	8	task difficulty
if_var19	num	8	spare
if_var20	num	8	spare
if_var21	num	8	spare
if_var22	num	8	spare
if_var23	num	8	spare
if_var24	num	8	spare
if_var25	num	8	spare

index OG202, key link6

File OG203, instructional features database

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link14	str	6	incoming link from working IF intermediate file
if_abbr	str	8	variable name for IF
ifdesc	str	40	description of instructional feature
benefit	num	8	benefit weight
link17	str	6	link to fidelity intermediate file
link18	str	6	link to working fidelity intermediate file
link19	str	6	link to components intermediate file

index OG203, key is link14

File OG210 fidelity intermediate file

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
source	str	2	source module code
linkin	str	6	incoming link from functions/cue/IF databases
intlink	str	6	internal link to next in chain
link8	str	6	outgoing link to fidelity database
index file OG210, key is source + linkin			

File OG211 data dictionary for variable name versus field names

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
rulebase	char	8	name of rule base
fldname	char	8	name of field
varname	char	8	corresponding name of variable
vardesc	char	50	gives description of parameter for screen preview
vartype	char	70	gives menu selections for screen preview
index file OG211, key is rulebase			

File OG212 fidelity database

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link8	str	6	incoming link from fidelity intermediate file
rulebase	str	8	name of rule base to use for this item
fi_var1	num	8	motion cue -- sway
fi_var2	num	8	motion cue -- heave
fi_var3	num	8	motion cue -- surge
fi_var4	num	8	motion cue -- yaw
fi_var5	num	8	motion cue -- pitch
fi_var6	num	8	motion cue -- roll
fi_var7	num	8	uncorrelated cues -- sway
fi_var8	num	8	uncorrelated cues -- heave
fi_var9	num	8	uncorrelated cues -- surge
fi_var10	num	8	uncorrelated cues -- yaw
fi_var11	num	8	uncorrelated cues -- pitch
fi_var12	num	8	uncorrelated cues -- roll
fi_var13	num	8	emergency cues -- sway
fi_var14	num	8	emergency cues -- heave
fi_var15	num	8	emergency cues -- surge
fi_var16	num	8	emergency cues -- yaw
fi_var17	num	8	emergency cues -- pitch
fi_var18	num	8	emergency cues -- roll
fi_var19	num	8	emergency tasks involved
fi_var20	num	8	prolonged accelerations
fi_var21	num	8	continuous control movement
fi_var22	num	8	spare
fi_var23	num	8	spare
fi_var24	num	8	spare
fi_var25	num	8	spare

fi_var26	num	8	spare
fi_var27	num	8	spare
fi_var28	num	8	spare
fi_var29	num	8	spare
fi_var30	num	8	spare
fi_var31	num	8	spare
fi_var32	num	8	spare
fi_var33	num	8	spare
fi_var34	num	8	spare
fi_var35	num	8	spare
fi_var36	num	8	spare
fi_var37	num	8	spare
fi_var38	num	8	spare
fi_var39	num	8	spare
fi_var40	num	8	spare
fi_var41	num	8	spare
fi_var42	num	8	spare
fi_var43	num	8	spare
fi_var44	num	8	spare
fi_var45	num	8	spare
fi_var46	num	8	spare
fi_var47	num	8	spare
fi_var48	num	8	spare
fi_var49	num	8	spare
fi_var50	num	8	spare

index OG212, key link8

File OG213, fidelity database

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link20	str	6	incoming link from working fidelity intermediate file
fi_abbrev	str	8	variable name for fidelity feature
fidesc	str	40	description of fidelity feature
techperf	num	8	tech performance
link21	str	6	link to component intermediate file
fidcrit	num	8	criticality of this fidelity item

index OG213, key is link20

File OG220, component intermediate file

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
source	str	2	incoming source module number
linkin	str	6	incoming link from earlier modules
intlink	str	6	internal link to next in chain
link10	str	6	outgoing link to component file

index OG220, key is source + linkin

File OG221, component data file

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link10	str	6	incoming link from component intermediate file
family	str	8	family name of this component
compdesc	str	40	description of component
link23	str	6	link to system intermediate file
rdadj	num	8	r&d adjustment factor
rdlow	num	8	r&d cost low estimate
rdexp	num	8	r&d most likely cost
rdhigh	num	8	r&d cost high estimate
invadj	num	8	investment cost adjustment factor
invlow	num	8	investment cost low estimate
invexp	num	8	investment most likely cost
invhigh	num	8	investment cost high estimate
fldadj	num	8	fielding cost adjustment factor
fldlow	num	8	fielding cost low estimate
fldexp	num	8	fielding most likely cost
fldhigh	num	8	fielding cost high estimate
conadj	num	8	MILCON cost adjustment factor
conlow	num	8	MILCON cost low estimate
conexp	num	8	MILCON most likely cost
conhigh	num	8	MILCON cost high estimate
susadj	num	8	sustainment cost adjustment factor
suslow	num	8	sustainment cost low estimate
susexp	num	8	sustainment most likely cost
sushigh	num	8	sustainment cost high estimate
techrisk	num	8	component technical risk
skedadj	num	8	R&D schedule adjustment factor
skedlow	num	8	R&D schedule low estimate
skedexp	num	8	R&D schedule most likely time
skedhigh	num	8	R&D schedule high estimate
pltf	num	8	provided level of technical support
util	num	8	component utilization

index OG221, key is link10
index OG222, key is family

File OG310 System intermediate file

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link23	char	6	incoming link from component file
intlink	char	6	internal link to next in chain
link24	char	6	outgoing link to system

File OG311 System file

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
link24	char	6	incoming link from system intermediate file
sysdesc	char	40	system description
rdadj	num	8	r&d adjustment factor
rdlow	num	8	r&d cost low estimate

rdexp	num	8	r&d most likely cost
rdhigh	num	8	r&d cost high estimate
invadj	num	8	investment cost adjustment factor
invlow	num	8	investment cost low estimate
invexp	num	8	investment most likely cost
invhigh	num	8	investment cost high estimate
fldadj	num	8	fielding cost adjustment factor
fldlow	num	8	fielding cost low estimate
fldexp	num	8	fielding most likely cost
fldhigh	num	8	fielding cost high estimate
conadj	num	8	MILCON cost adjustment factor
conlow	num	8	MILCON cost low estimate
conexp	num	8	MILCON most likely cost
conhigh	num	8	MILCON cost high estimate
susadj	num	8	sustainment cost adjustment factor
suslow	num	8	sustainment cost low estimate
susexp	num	8	sustainment most likely cost
sushigh	num	8	sustainment cost high estimate
techrisk	num	8	component technical risk
skedadj	num	8	R&D schedule adjustment factor
skedlow	num	8	R&D schedule low estimate
skedexp	num	8	R&D schedule most likely time
skedhigh	num	8	R&D schedule high estimate
nodev	num	8	number of devices needed per year
chreq	num	8	classroom hours required by system
dhreq	num	8	device hours required by system
shreq	num	8	setup hours required by system
susaving	num	8	setup savings provided by system
util	num	8	system utilization
tlach	num	8	training level achieved by system
ttotr	num	8	time to train to required level

File OG400 Outline file alias outline

<u>FIELD</u>	<u>TYPE</u>	<u>NUM</u>	<u>DESCRIPTION</u>
para	char	10	paragraph number in outline
serial	char	21	(date(8)+time(8)+module(1)+function(1)+line(3))
title	char	40	title of outline section
modules	char	30	modules/functions putting notes in this section

index file OG400, key is para + serial

Working Files Identification & Description

- File CFPNN100 - Identical structure to OG104, contains selected functions
- File CFPNN110 - Identical structure to OG111, contains selected cues
- File CFPNN111 - Identical structure to OG0C, contains backwards links to functions from cues
- File CFPNN120 - Identical structure to OG120, contains selected simulation checklist items
- File CFPNN200 - Identical structure to OG200, functions/cues to IF data working intermediate file
- File CFPNN201 - Identical structure to OG200, functions/cues to selected IFs intermediate file
- File CFPNN202 - Identical structure to OG202, contains IF data for each selection
- File CFPNN203 - Identical structure to OG203, contains selected instructional features
- File CFPNN204 - Identical structure to OG0C, contains backwards links to functions and cues
- File CFPNN210 - Identical structure to OG210, functions/cues/IF to fidelity data working intermediate file
- File CFPNN211 - Identical structure to OG210, functions/cues/IF to selected fidelity items intermediate file
- File CFPNN212 - Identical structure to OG212, contains fidelity data for each selection
- File CFPNN213 - Identical structure to OG213, contains selected fidelity features
- File CFPNN214 - Identical structure to OG0C, contains backwards links to functions, cues and IFs
- File CFPNN220 - Identical structure to OG221, contains selected components. *Indexes CFPNN220, key is link10; CFPNN221, key is family*
- File CFPNN222 - Identical structure to OG0C, contains backwards links to earlier modules
- File CFPNN310 - Identical structure to OG310, system

intermediate file. Index CFPNN310, key is
link23

File CFPNN311 - Identical structure to OG311, system file.
Index CFPNN311, key is link24

File CFPNN312 - Identical structure to OG0C, backwards
intermediate file. Index CFPNN312, key is
source (always "22")+blinkin

File CFPNN400 - Identical structure to OG400, contains current
outline

APPENDIX C

Rule Base Development Guidance

Developing instructional feature or fidelity rule bases requires some knowledge of the internal workings of the CFP-Aid. This appendix briefly describes some of the requirements for developing rule bases and integrating them into the CFP-Aid.

Rule Base Requirements

There are some minor differences between the two kinds of rule bases. These differences are due mainly to differences in the links between fidelity features, instructional features, and other elements of the CFP-Aid; and to differences in the initial rule bases that were incorporated into the aid.

Goal variable must be "I_F" for instructional feature rule bases and "FIDELITY" for fidelity rule bases. These variables should be set to true when the rule base is completed and some instructional features or fidelity features have been selected.

There must be 50 or fewer input variables for fidelity rule bases and 25 for instructional feature rule bases. The limits come from the number of fields in OG212.ITB and OG202.ITB, respectively. If a rule base requires more than the limit, we recommend breaking the rule base into smaller rule bases.

Keep in mind that when the rule base is run, some of the variables may be known, and some unknown. Furthermore, there will often be no way to guarantee that the incoming data will be internally consistent or meaningful. Consequently, it is necessary to write variable sections so that they will stand alone. That is, don't assume that other variables will be known when a query is made for a certain variable. Set e.ocon to true and clear the screen before every data request. Clear the screen again and set e.ocon to false after the request. Don't rely on the specific order that questions will be asked unless you can guarantee that the questions will be asked in that order.

It may be necessary to write an initialization section to weed out inconsistent combinations of data values and perform some appropriate action. For example, in a multiple choice menu, it is generally not the case that some variables will evaluate to "true" while others will remain unknown. However, this combination of values can easily arrive to the rule base as input from the database. To avoid this possibility, it is necessary to write an initialization section that compares the values of variables that are assessed in a single menu. If one of these variables is unknown, then all should be set to unknown.

GURU fuzzy variables should be avoided. If they must be used, then the fuzzy variables in the rule base must be represented by several variables outside of the rule base (one for each possible value of the fuzzy variable). The initialization section must contain code that translates the variables in the database into the fuzzy variable in the rule base. The reverse translation must be done in the variables section for the fuzzy variable.

It is also advisable to avoid numerical input variables, if possible. Under the current system, numerical values can only be edited in the rule base, consequently leading to more difficulty in using the rule base and maintaining its data.

Input variables are defined in the appropriate data dictionary, OG201.ITB for instructional features, and OG211.ITB for fidelity. For example, the following define a variable named "hm-sway" that is the first variable ("fi_var1") in the fidelity rule base, "motion."

```
rulebase = "motion"
fldname = "fi_var1"
varname = "hm_sway"
vardesc = "Are there any high or moderate sway cues?"
vartype = "unknown no yes"
```

the field "vardesc" refers to the question that is used to determine the value of the variable, while "vartype" gives a menu of possible answers (with a space between answers). In the database, any value less than 0.0 signifies that the variable is unknown. The value 0.0 receives the second value in vartype ("no"); the value 1.0 receives the third value ("yes"); and so on, until all of the values are used up.

Instructional features and fidelity variable names should appear explicitly as outputs of the rules. These variables, as well as any other global variables, should be given names that do not conflict with any existing variable names or GURU keywords. The output variables of the rule bases are logical variables. There is one variable for each instructional feature or fidelity item. They are defined in OG203 for instructional features and OG213 for fidelity features, where they are also linked to components. These variables must be set to true if the fidelity item or instructional feature is indicated by the input data. Otherwise they are set to false.

Finally, try to make the user interface consistent with the rest of the program. Use the variable "g_menclr" to set the colors for menus. Menus should be in low intensity with double borders. Don't use or clear the top line of the display, which shows the task, cue, or instructional feature being examined by the rule base and the name of the rule base.

Relevant Databases

Each of the rule bases is directly associated with four system database files. The following discussion describes the information that must be in each of these databases to accommodate a new rule base.

Instructional features. The instructional features module addresses four system database files, OG200.ITB, OG201.ITB, OG202.ITB, and OG203.ITB. The following changes should be made in these databases to incorporate a new instructional feature rule base into the CFP-Aid.

OG200.ITB This file represents the links from tasks and cues to instructional feature rule base data. In general, this file is edited in function or cue edit routines. It must be edited to include links to records in OG202.ITB that reference the new rule base. Otherwise, the rule base will never be run, even if it is fully integrated into the CFP-Aid.

OG201.ITB This file contains the dictionary that defines the rule base variable names. It must be edited to define the variable names for any new rule base. The information in this database is also used to generate the menu for data confirmation in the "automatic with confirmation" execution of the rule base. This activity is conceptually part of instructional features edit routines, but is currently only partially implemented.

OG202.ITB This file contains input data used by the rule base. It is edited in the task or cue edit routines. It can also be edited in the instructional features "automatic with confirmation" menu. Since it is possible to answer questions directly in the rule base, it is possible to use the rule base to fill the records of this file, though it is not possible to use the rule base to create those records.

OG203.ITB This file links instructional features to fidelity data and components. Editing this file to represent the new rule base is required if the instructional features are to have any impact on the components chosen to be in a training system. The editing should be accomplished with the instructional feature edit procedure.

Fidelity features. The fidelity data files are directly analogous to instructional feature files, except that fidelity has somewhat different links from those of instructional features. The fidelity module addresses four system database files, OG210.ITB, OG211.ITB, OG212.ITB, and OG213.ITB. The following changes should be made in these databases to

incorporate a new instructional feature rule base into the CFP-Aid.

OG210.ITB This file represents the links from tasks, cues, and instructional features to fidelity rule base data. In general, this file is edited in function, cue, or instructional feature edit routines. It must be edited to include links to records in OG212.ITB that reference the new rule base.

OG211.ITB This file contains the dictionary that defines the fidelity rule base variable names. It must be edited to define the variable names for any new rule base.

OG212.ITB This file contains input data used by the fidelity rule base. It is edited in the task, cue, or instructional feature edit routines. It should be noted that current capabilities for editing rule base data do not allow for editing numerical data except using the rule base in "answer questions" mode.

OG213.ITB This file links fidelity features to components.